

TUTORIAL

LaTeX tutorial for the standardization and automation of population analysis reports

Thomas Frank | Silke Gastine | Klaus Lindauer | Heiner Speth | Ashley Strougo | Andreas Kovar

Sanofi-Aventis Deutschland GmbH,
R&D, Frankfurt, Germany

Correspondence

Thomas Frank, Industriepark Höchst,
Bldg. H831, 65926 Frankfurt a.M.,
Germany.
Email: thomas.frank@sanofi.com

Funding information

No funding was received for this work.

Abstract

Writing population analysis reports that fulfilling the specifications of submission readiness is a time-consuming process and prone to human error. Thus, there is a need to streamline the creation of these reports through elements of standardization and automation. LaTeX is considered a suitable program package capable of creating long, modular, structured documents, and, because of its typographic quality, includes formula typesetting. The presented automation scripts together with easily adjustable LaTeX templates are designed to enable the reader to understand and reproduce a typical workflow from analysis to reporting. The focus of this tutorial is to use an example of a population pharmacokinetic analysis to show how to work with the proposed automated structures allowing even a reader new to the concept of LaTeX to automatize the reporting workflow and customize the templates for their specific needs.

INTRODUCTION

One of the main purposes of a population analysis report is to provide detailed documentation of analytical methods and conduct, making analyses reproducible and facilitating internal and/or regulatory review.¹ The expected content and level of detail of the individual components of a population analysis report are essentially identical in the different regions of the world, as the respective guidelines^{2–4} aim to promote uniformity and standardization of reporting from a regulatory perspective. A head-to-head comparison of the expected content in each section of a population pharmacokinetic (PopPK) study report across different regions is shown in Supplementary Material S1.

Nevertheless, producing reports that meet submission readiness specifications is a time-consuming process that is prone to human error because many steps are repetitive and require quality assurance measures. Standardization and elements of automation should thus be used for recurring steps as much as possible. Typically, a project-centric model development workflow is applied, where analyses are roughly divided into data, analysis, and report subsections through an appropriate directory and file structure.^{5,6} However, the need to make this workflow more automated and reproducible is often interrupted by the fact that the actual report generation is performed separately from the analysis using a standard word processor. Additional problems encountered are copy-pasting errors across documents, manual checking of acronym introductions, or

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2021 The Authors. *CPT: Pharmacometrics & Systems Pharmacology* published by Wiley Periodicals LLC on behalf of American Society for Clinical Pharmacology and Therapeutics

proper cross-referencing. Back-tracing of up-to-date tables and figures can also become cumbersome when not working in a predefined and automated environment. These gaps can be closed by using a programming language, which already allows for elements of standardization and automation. TeX, a programming language for typesetting, is the centerpiece of the program package LaTeX,⁷ which is the de facto standard document preparation system used for the communication and publication of scientific documents. The program easily masters the creation of long, modular, structured documents with unmatched typographic quality, including typesetting of formulas, which are all aspects relevant to population analysis reports. Admittedly, for authors who are used to creating reports with word processors, switching to a markup language like LaTeX can be quite a hurdle. In order to lower the entry threshold for new LaTeX users, efforts have recently been made to provide templates and scripts specifically adapted for authors in the field of pharmacometrics.⁵ What is still missing, however, is an example of how to automatize getting from the initial data via the analysis to the submission ready report. This tutorial attempts to fill this gap by providing concrete examples of code through a step-by-step description using a typical pharmacometric example taken from the literature.

As a case study, a PopPK analysis using the nonlinear mixed-effects modeling software NONMEM, which was the subject of an earlier tutorial,⁸ is re-created for this purpose. Furthermore, the processing of a NONMEM analysis, including generation of tables and figures (post-processing) and their integration into a LaTeX report using automated directory structures is shown.

To follow the steps described in this tutorial, it is not necessary to execute NONMEM or create the desired tables and figures from the NONMEM runs in R, because all results are provided. All files are available in Supplementary Materials S2 and additionally on an open GitHub (https://github.com/tfrank1971/LaTeX_Tutorial_CPT_PSP).

CASE STUDY: A SIMPLE EXAMPLE OF A POPULATION PHARMACOKINETIC ANALYSIS

To illustrate how to work with LaTeX and the automated structures, the pharmacokinetic (PK) model No. 504 from the NONMEM tutorial part 1 by R. Bauer⁸ was executed and analyzed. A full example report of this case study is presented in the Supplementary Material S2 (file *sample.pdf*).

In this NONMEM example, 60 surgical patients in a phase IIa study received a 1 mg, 2 mg, or 4 mg dose of an anesthetic as a 5-h constant-rate intravenous infusion. PK samples were obtained at four time points after the start of the infusion (1, 5, 12, and 24 h). Patient demographic data (weight, age, and sex) were used to

determine the covariate relationships with the PK parameters. Data were fitted to a one-compartment constant-rate infusion model parametrized in terms of clearance (CL) and volume of distribution (V). The goal of the example report was to describe the evolution of this model from the pharmacostatistical model via the covariate model to the final PopPK model in a comprehensible way by means of informative tables and figures.

The analysis was performed using NONMEM software version 7.4.1 (ICON, Development Solutions, Elliot City, MD, USA) running under openSUSE Linux (x86_64, Leap 15.1) on a computer cluster.⁹ On this cluster, NONMEM jobs are queued using Maui and Torque.¹⁰ The results of the NONMEM runs are returned in archive file format (*.tar.gz) with unique archive numbers assigned by the queueing system.

Due to every NONMEM user having their preferred individual environment setup, and thus to allow reproducibility in this tutorial, the respective run archives are supplied in Supplementary Material S2.

The base model is archived in the compressed file *run504_no_cov.1117171.tar.gz*. Here, interindividual variability (IIV) is estimated for both CL and V. The residual unexplained variability is included via a proportional error, as the drug assay had a constant coefficient of variation. The full covariate model corresponding to model 504.c_{tl}⁸ is archived in file *run504_full_cov.1117172.tar.gz*, and the final model corresponding to model 504f.c_{tl},⁸ where the covariate modeling is simplified by introducing fixed allometric exponents, is archived in file *run504_with_cov_fix.1117173.tar.gz*. Graphical analysis, descriptive statistics, and the evaluation of NONMEM outputs were conducted using R. In line with reporting guidelines, the used R scripts were documented in the Appendix of the sample report in Supplementary Material S2 (file *sample.pdf*).

The general workflow to obtain the sample report was:

- Create directory structure using Perl script *makereprt.pl*,
- Execute necessary NONMEM runs,
- Create desired tables and figures from the NONMEM runs in R,
- Edit the LaTeX template by inserting the appropriate content into each section,
- Compile the final report using *latexmk*.

PREREQUISITES

- Linux (openSUSE Linux Leap 15.1 and above),
- TeX Live 2020 or later (www.tug.org/texlive/), full installation recommended, usually integrated under Linux,
- Perl (Perl 5, version 26, subversion 1 [v 5.26.1] or higher), usually integrated under Linux,
- Perl script *extractrep.pl* (optional, for users who want to repeat the analysis),

- R version 3.6.1 or higher (www.r-project.org; optional, for users who want to repeat the analysis),
- Several R libraries and their dependencies (Appendix C.2 in file *sample.pdf*, Supplementary Material S2; optional, for users who want to repeat the analysis),
- LaTeX code editor (e.g., Kile or TeXstudio; www.textstudio.org),
- pdftk (version 2.02 or higher; optional, for users who want to repeat the analysis),
- Microsoft standard fonts.

The example was also successfully reproduced on Ubuntu 20.04 installed in the Windows Subsystem for Linux (WSL2), on Windows 10 Professional. This required only minor modification once TeX Live, latexmk, and other dependencies were in place. In addition, Microsoft Core Fonts should be installed via the Ubuntu package *ttf-mscorefonts-installer* before compilation using this setup.

INSTALLATION OF THE AUTOMATED DIRECTORY STRUCTURE AND TEMPLATE

Here, we describe how the supplied files are used to obtain the automated directories and how to make the best use of the LaTeX template through integration with R scripts and NONMEM output.

Predefined working environment introduces standardization

An overview of the supplied files (LaTeX source code, analysis dataset, NONMEM output, and scripts) is given in Figure 1. The execution of the supplied Perl script *makereprt.pl* creates a directory structure with main subdirectories: data, analysis, and report. The NONMEM input data file can be dropped into subdirectory “data” (and can be linked into “analysis” subdirectory via symbolic link). The NONMEM runs are then executed in subdirectory “analysis”; subdirectory “nmnqs” contains the results from the NONMEM runs returned from a queueing system as compressed (*.tar.gz) files. Here, the control stream has the extension *.con and the report file *.rep. Depending on the user’s setup, adaptations can be made to adequately address the NONMEM output.

Generic R scripts are used for post-processing

R scripts to create plots and tables are also automatically delivered using the above-mentioned Perl script. These

are also automatically dropped into respective “plots” and “tables” subdirectories. Script *exploratory.R* summarizes the NONMEM input data set (e.g., samples and subjects), script *postprocessing.R* and its dependent script *standard.plots.R* document the creation of goodness-of-fit plots and parameter summary tables. The code of further R functions included in the above R scripts is also provided in the subdirectory “analysis/.R.” For repeated use, all R functions need to be loaded in the R workspace. In addition, the pdftk program and the Perl script *extractrep.pl* need to be installed (e.g., in /home/user/bin or /usr/local/bin; check possible path variables in the command line interface [shell] using the command “echo \$PATH”).

The raw LaTeX template is supplied in subdirectory “report.” It has a modular structure, as shown in Table 1 and Figure 1. This way, files holding actual content (subdirectory “textblocks”) are separated from structure-defining files. The content files include LaTeX chunks for each respective section of the report and need to be filled with content to match the user’s needs. As soon as the report has been compiled, everything is wrapped into one PDF file. File *sample.pdf* in Supplementary Materials S2 provides a first impression of the layout.

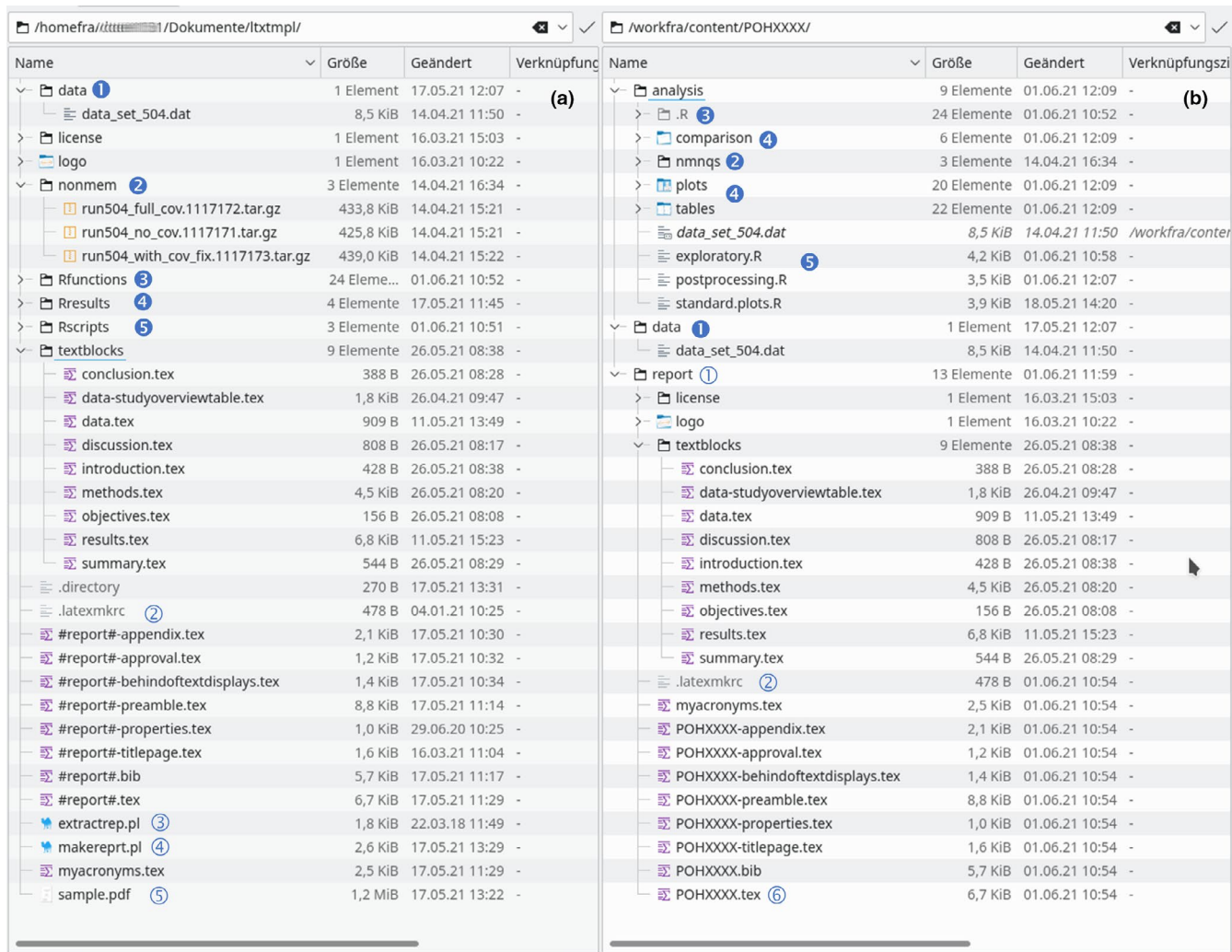
It is recommended to extract the zip file containing all above-mentioned files (Supplementary Materials S2) first to a local directory. As detailed below, this directory should be kept separate from the actual working directory.

Automation enables recycling of repetitive parts of reports

For reproducibility, maintaining a common directory structure across different projects is beneficial. Repeated creation, however, can become cumbersome. Therefore, Perl script *makereprt.pl* provided in Supplementary Materials S2 was created. To make best use of the Perl script, certain steps are necessary, as summarized in Table 2. First, move the file *makereprt.pl* to /usr/local/bin or similar (e.g., /home/user/bin; check possible path variables in the shell using the command “echo \$PATH”). Next, the Perl script needs to be personalized. Open *makereprt.pl* with a text editor and change line 9 to point to your own directory:

```
$LATEXTMPLDIR = "/home/user/documents/ltxtmpl"; #this is the source directory of the template
```

Once these preparatory steps have been accomplished, the Perl script can be executed from the shell. By navigating to the designated top directory and executing the Perl script with the following command, the new project will be created therein.



Name	Größe	Geändert	Verknüpfung	Name	Größe	Geändert	Verknüpfung
data (1)	1 Element	17.05.21 12:07	-	analysis	9 Elemente	01.06.21 12:09	-
data_set_504.dat	8,5 KiB	14.04.21 11:50	-	.R (3)	24 Elemente	01.06.21 10:52	-
license	1 Element	16.03.21 15:03	-	comparison (4)	6 Elemente	01.06.21 12:09	-
logo	1 Element	16.03.21 10:22	-	nmnqs (2)	3 Elemente	14.04.21 16:34	-
nonmem	3 Elemente	14.04.21 16:34	-	plots (4)	20 Elemente	01.06.21 12:09	-
run504_full_cov.1117172.tar.gz	433,8 KiB	14.04.21 15:21	-	tables	22 Elemente	01.06.21 12:09	-
run504_no_cov.1117171.tar.gz	425,8 KiB	14.04.21 15:21	-	data_set_504.dat	8,5 KiB	14.04.21 11:50	/workfra/conten
run504_with_cov_fix.1117173.tar.gz	439,0 KiB	14.04.21 15:22	-	exploratory.R (5)	4,2 KiB	01.06.21 10:58	-
rfuctions (3)	24 Eleme...	01.06.21 10:52	-	postprocessing.R	3,5 KiB	01.06.21 12:07	-
Rresults (4)	4 Elemente	17.05.21 11:45	-	standard.plots.R	3,9 KiB	18.05.21 14:20	-
Rscripts (5)	3 Elemente	01.06.21 10:51	-	data (1)	1 Element	17.05.21 12:07	-
textblocks	9 Elemente	26.05.21 08:38	-	data_set_504.dat	8,5 KiB	14.04.21 11:50	-
conclusion.tex	388 B	26.05.21 08:28	-	report (1)	13 Elemente	01.06.21 11:59	-
data-studyoverviewtable.tex	1,8 KiB	26.04.21 09:47	-	license	1 Element	16.03.21 15:03	-
data.tex	909 B	11.05.21 13:49	-	logo	1 Element	16.03.21 10:22	-
discussion.tex	808 B	26.05.21 08:17	-	textblocks	9 Elemente	26.05.21 08:38	-
introduction.tex	428 B	26.05.21 08:38	-	conclusion.tex	388 B	26.05.21 08:28	-
methods.tex	4,5 KiB	26.05.21 08:20	-	data-studyoverviewtable.tex	1,8 KiB	26.04.21 09:47	-
objectives.tex	156 B	26.05.21 08:08	-	data.tex	909 B	11.05.21 13:49	-
results.tex	6,8 KiB	11.05.21 15:23	-	discussion.tex	808 B	26.05.21 08:17	-
summary.tex	544 B	26.05.21 08:29	-	introduction.tex	428 B	26.05.21 08:38	-
.directory	270 B	17.05.21 13:31	-	methods.tex	4,5 KiB	26.05.21 08:20	-
.latexmkrc (2)	478 B	04.01.21 10:25	-	objectives.tex	156 B	26.05.21 08:08	-
#report#-appendix.tex	2,1 KiB	17.05.21 10:30	-	results.tex	6,8 KiB	11.05.21 15:23	-
#report#-approval.tex	1,2 KiB	17.05.21 10:32	-	summary.tex	544 B	26.05.21 08:29	-
#report#-behindoftextdisplays.tex	1,4 KiB	17.05.21 10:34	-	.latexmkrc (2)	478 B	01.06.21 10:54	-
#report#-preamble.tex	8,8 KiB	17.05.21 11:14	-	myacronyms.tex	2,5 KiB	01.06.21 10:54	-
#report#-properties.tex	1,0 KiB	29.06.20 10:25	-	POHXXX-appendix.tex	2,1 KiB	01.06.21 10:54	-
#report#-titlepage.tex	1,6 KiB	16.03.21 11:04	-	POHXXX-approval.tex	1,2 KiB	01.06.21 10:54	-
#report#.bib	5,7 KiB	17.05.21 11:17	-	POHXXX-behindoftextdisplays.tex	1,4 KiB	01.06.21 10:54	-
#report#.tex	6,7 KiB	17.05.21 11:29	-	POHXXX-preamble.tex	8,8 KiB	01.06.21 10:54	-
extractrep.pl (3)	1,8 KiB	22.03.18 11:49	-	POHXXX-properties.tex	1,0 KiB	01.06.21 10:54	-
makereprt.pl (4)	2,6 KiB	17.05.21 13:29	-	POHXXX-titlepage.tex	1,6 KiB	01.06.21 10:54	-
myacronyms.tex	2,5 KiB	17.05.21 11:29	-	POHXXX.bib	5,7 KiB	01.06.21 10:54	-
sample.pdf (5)	1,2 MiB	17.05.21 13:22	-	POHXXX.tex (6)	6,7 KiB	01.06.21 10:54	-

FIGURE 1 Structure of the source and target directory of the LaTeX template with associated files. The source directory is shown in the left field (a), and the target directory with the three main subdirectories “analysis,” “data,” and “report” in the right field (b). The Perl script *makereprt.pl* (4) creates the target directory structure and sorts various files under it. This Perl script should be moved to a directory containing executables (e.g., */usr/local/bin/*). The NONMEM input data file is located in subdirectory “data” (1). The NONMEM results, returned as *.tar.gz files, are put under “nmnqs” (2) to enable the processing by the R script *postprocessing.R*. Additional “helper” functions are copied to the hidden directory “.R” (3). The results of R processing are placed in subdirectory “plots,” “tables,” and “comparison” (4), and the R scripts are located directly in subdirectory “analysis” (5). The LaTeX source files of the report are placed in subdirectory “report” (1) with files containing content of the respective sections being placed in subdirectory “textblocks.” The hidden file *.latexmkrc* is the configuration file for automated compilation by *latexmk* (2). The Perl script *extractrep.pl* (3) completes the action of some of the R functions in “.R” and should be moved to a directory containing executables (e.g., */usr/local/bin/*). Note that through the action of *makereprt.pl* (4) the string “#report#” inside the LaTeX source files and in the file names is replaced by the argument of the function call (in the present example “POHXXX”). File *POHXXX.tex* (6) is the root or master file since all other TeX files are included here. A pre-compiled sample report is provided in file “sample.pdf” (5).

```
makereprt.pl POHXXX
```

When the script is invoked, it copies the LaTeX source files along with what is needed for post-processing and compilation from the script’s source directory and places them in subdirectories. At the same time, the Perl script renames the LaTeX source files and some variables in them with an argument (a string) specified by the user.

In the supplied example, the project is called POHXXX (Figure 1), which will also be used for the rest of the tutorial. The project name is also passed to items created in the directory structure of the new project (e.g., file *POHXXX-properties.tex*). Note that this string is not allowed to have characters like ‘:’, ‘-’, ‘_’, or spaces. This way, a specific project code is anchored in the file name and later serves as a document reference in the PDF file.

TABLE 1 Overview of LaTeX template files (Supplementary Materials S2)

Name	Type	Description
Data	Directory	Contains NONMEM input data set <i>data_set_504.dat</i>
License	Directory	Contains a text file holding GNU AFFERO GENERAL PUBLIC LICENSE
Logo	Directory	Contains image file <i>logo.png</i>
NONMEM	Directory	Contains 3 compressed (*.tar.gz) files containing NONMEM control streams and results for base model (No. 1117171), full covariate model (No. 1117172) and final model (No. 1117173)
Rfunctions	Directory	Contains 24 R functions needed to post-process NONMEM archives
Results	Directory	Includes subdirectories containing tables and plots belonging to the NONMEM analysis (output created by the 3 R scripts <i>standard.plots.R</i> , <i>exploratory.R</i> , and <i>postprocessing.R</i>)
Rscripts	Directory	Contains 3 R scripts <i>standard.plots.R</i> (functions to create goodness-of-fit plots and summary tables for NONMEM runs), <i>exploratory.R</i> (descriptive statistics of PK samples and subjects, concentration versus time plots of raw data), and <i>postprocessing.R</i> (post-processing steps for NONMEM runs)
Textblocks	Directory	Contains LaTeX source files that hold the actual content of the report (e.g., <i>summary.tex</i> , <i>introduction.tex</i> , etc.)
.latexmkrc	File	(Hidden) configuration file for automated report compilation with <i>latexmk</i>
#report#.bib	File	Literature database file from which the reference list will be created by BibLaTeX and biber
#report#.tex	File	The main source file (document environment) in which the titlepage, approval page, textblocks, behind-of-text displays, tables, and appendix get included
#report#-appendix.tex	File	Source file of the appendix
#report#-approval.tex	File	Source file of the approval page
#report#-behindoftextdisplays.tex	File	Source file to include additional figures and tables behind the core sections of the report
#report#-preamble.tex	File	Preamble (e.g., to define common macros to plot figures and tables, to define layout, fonts, page header, and footer)
#report#-properties.tex	File	Defines the meta information (i.e., title, subtitle, author name, reviewer name, and other variables used by the other template files)
#report#-titlepage.tex	File	Source file of the titlepage
extractrep.pl	File	Perl script completing the action of some of the R functions under “Rfunctions”
makereprt.pl	File	Perl script that retrieves the template files from a directory and renames them according to user input. Adapt the variable $\$LATEXTPMLDIR$ as explained in the text.
myacronyms.tex	File	Definition of terms and symbols; used to create the list of acronyms and definition of terms
sample.pdf	File	Example of the compiled report template

If necessary, the directory structure can be individualized and adapted to specific needs and requirements. In this case, the file *makereprt.pl* has to be edited accordingly. For example, the *rsync* part of the Perl script (lines 80–89) can be extended to provide further document templates to customize the preparation of the analysis.

The following section covers the general structure of the template, the report sections, where to write the content, how to create cross references, and how to manage the layout of the report. For general LaTeX commands (e.g., writing symbols, special characters, etc.), please refer to the paper by Lammarsch and Schubert.¹¹

WORKING WITH THE TEMPLATE AND ADAPTING IT TO YOUR INDIVIDUAL NEEDS

The template has a modular structure and thus separates content from layout as far as possible by delivering several LaTeX source files (Table 1 and Figure 1). File *POHXXXX.tex* is the root or master document of the report. It contains the document class and document environment in which all other components of the report are included. These are recognized by their file names. The actual content is prepared in separate files located in the subdirectory /textblocks. All content files are included into the report

TABLE 2 Steps to install the template (Supplementary Materials S2)

Step	Description
1	Unpack the zip file in a directory of choice
2	Move file <i>makereprt.pl</i> to /usr/local/bin or similar (to check with “echo \$PATH”)
3	Move file <i>extractrep.pl</i> to /usr/local/bin or similar (to check with “echo \$PATH”; optional, only needed if analysis is to be repeated)
4	Edit variable \$LATEXMPDIR in file <i>makereprt.pl</i> to point to the source directory of the template (see step 1)
5	Call “makereprt.pl POHXXXX” in the command line interface (shell) at the position where the report should be created (keep directory disjunct from step 1)

via the LaTeX `\input` command. This is exemplified with the following lines of LaTeX code (file *POHXXXX.tex*, lines 132 to 135):

```
\section{INTRODUCTION}
\label{sec:introduction}
\input{../textblocks/introduction}
```

Note that the `\input` command does not require the file suffix `*.tex`. Further content files can be created in the directory “textblocks” when necessary. The document class used in this tutorial is based on the KOMA-Script package¹² and also supports European formats. Further guidance on how to change the reports general appearance is described in Supplementary Materials S3.

Document properties

Because the document preamble of the template is considerably long, it is kept separately in the *POHXXXX-preamble.tex* file. The preamble serves several purposes. First, add-on packages performing typographical tasks not directly supported by base LaTeX are loaded (command `\usepackage`). Additionally, it contains the definition of macros used to insert figures and tables (lines 81 to 220). Settings for the bibliography (biblatex package) and for the list of abbreviations (glossaries package) are defined in lines 45 to 47 and lines 52 to 58, respectively. Further information is given in the following sections and in Supplementary Materials S3.

Document properties or meta information of a report are variables that describe basic characteristics of the report (file *POHXXXX-properties.tex*). The `\newcommand` command defines new variables, with the first argument being the name and the second the value of the variable. These variables are used to populate information on the title page, signature (approval) page, and page headers. Some variables, such as the project name, are determined by the Perl script *makerprt.pl*. All items marked with `<...>` need to be adapted manually for a new project.

Title page

The title page is the first page of the report and should include all the information necessary for a reader to identify the contents of the report, such as author(s), origin of the report, and report type. Key information on the title page, such as author and reviewer name, institution, and job title, are already defined in file *POHXXXX-properties.tex*. Advanced users can make layout adjustments by adapting the title page environment defined in file *POHXXXX-titlepage.tex*. To change the logo, replace the image file provided in the /logo subdirectory.

Approval page

The approval page lists the author/s, reviewer/s, and approver/s of the report together with a confirmatory statement (file *POHXXXX-approval.tex*). The actual content has already been defined by variables in file *POHXXXX-properties.tex*. This template assigns one person each to act as author, reviewer, and approver. There might be more people involved in writing or reviewing the report. To add additional authors or reviewers, an extra code block needs to be added corresponding to lines 13 to 23 in file *POHXXXX-approval.tex*. In general, the approval page is optional and added when the report is submitted to an electronic signature workflow. This is usually the case when using document management systems (e.g., Veeva Vault RIM; Veeva Systems Inc., Pleasanton, CA, USA). Alternatively, the page can also be printed out, signed-off manually, and imported as a PDF file. To do so, comment out line 67 in file *POHXXXX.tex* and uncomment line 68. In this case, the statements “Total number of pages: 66 + 1 Page for Electronic Signatures” (file *POHXXXX-titlepage.tex*, line 27) and “signed electronically” (file *POHXXXX-approval.tex*) also need to be adapted.

Table of contents, list of tables, and list of figures

LaTeX creates the table of contents, list of tables, and list of figures each with a single command: `\tableofcontents`,

\listotables, and \listoffigures (file *POHXXXX.tex*, lines 100 to 111). The depth of the table of contents is controlled in lines 62 to 63 of the preamble.

In standard LaTeX, section headings and page numbers are taken from the last compilation cycle for a document, thus after applying changes to a new document, it must be compiled at least twice to get content tables correct. As described later, compilation with *latexmk* guarantees that LaTeX will be started as many times as necessary to create or update all tables of contents.

List of abbreviations and definition of terms

Throughout a report many acronyms are usually used. By using the glossaries package¹³ these acronyms, abbreviations, and definition of terms and symbols are managed making manually checking of the appropriate introduction redundant. All items are defined in file *myacronyms.tex* using the \newacronym command:

```
\newacronym[options]{<label>}{<abbrv>}
{<full>},
```

where <label> is the unique label identifying the acronym, <abbrv> is the abbreviated form of the acronym and <full> is the expanded text. Plural forms can be defined as an [option] (e.g., \newacronym[longplural={diagonal matrices}]{diagonal matrix}).

To add an acronym to the list of abbreviations, one of the commands shown in Table 3 can be inserted in the text. Here, the \gls command will display “<full> (<abbrv>).” To print the plural form of the abbreviation use \glspl{<abbrv>}. On subsequent use, only the

abbreviation will be displayed. More information can be found in ref. 14.

The key commands to create the list of abbreviations are defined in the preamble (file *POHXXXX-preamble.tex*, lines 52 to 58). The command \printglossary in the main document (file *POHXXXX.tex*, line 120) then prints the list of abbreviations.

Main report sections

Core text for objectives, methods, results, etc. are described in their designated text blocks. Subsections can be included to further organize the report. Text writing can be performed with a standard text editor or a LaTeX specific editor (see section “prerequisites”).

The report summary (synopsis) is placed in the abstract environment (\begin{abstract}... \end{abstract}, file *POHXXXX.tex*, lines 80 to 88). The text block of the abstract is usually tighter compared to the rest of the report. In the present template, it is widened to fit with general layout (\let\vfil\relax, line 79). If figures or tables are shown in the abstract, the counter may need to be reset. This can be achieved for a table with the command \setcounter{table}{0} (file *POHXXXX.tex*, line 128).

List of references

Managing the bibliography requires an external literature database file. A sample file is provided in *POHXXXX.bib*, including some predefined sample references. Referencing online sources in a report that is part of a submission document, the \nolinkurl command in the *.bib file needs to be considered, as it is usually not allowed to have activated

TABLE 3 Frequently used acronym commands

Command	Explanation
\gls{<label>}	Displays the full form the first time it is used and just the short form on subsequent use
\glspl{<label>}	Prints the plural of the defined term, other than that it behaves in the same way as gls
\glsadd{<label>}	Print entries in the glossary if not referenced in the text
\Gls{<label>}	Prints the singular form of the term with the first character converted to upper case
\Glspl{<label>}	Prints the plural form with first letter of the term converted to upper case
\acrshort{<label>}	To print the abbreviation “<abbrv>”
\acrfull{<label>}	Writes the spelled-out form of the acronym “<full> (<abbrv>)”
\aclong{<label>}	To print the long version of an acronym “<full>”
\acrshortpl{<label>}	To print the plural of abbreviation “<abbrv>”
\acrfullpl{<label>}	To write the spelled out plural form of the acronym “<full> (<abbrv>)”
\aclongpl{<label>}	To print the plural long version of an acronym “<full>”

Note: The general form of the acronym database entry is: \newacronym[options]{<label>}{<abbrv>}{<full>}.

hyperlinks in the reference list (file *POHXXXX.bib*, lines 152 to 156).

Further entry types are given by Lammarsch and Schubert.¹¹ To avoid manual creation of databases, many common reference manager packages, such as EndNote (Clarivate Analytics, Philadelphia, PA, USA), allow citations to be exported in native BibTeX format. Alternatively, references can be exported from PubMed via the interface TexMed.¹⁵

Citations in text are placed with the command `\autocite{Comets2008}` (first argument after `@Article` in *.bib* file). To combine several citations, a comma-separated list of bib-file entries can be used: `\autocite{Comets2008, Nguyen2012,...}`.

The biblatex package¹⁶ is commonly used by LaTeX users and is therefore implemented in the template. Biblatex provides a powerful and flexible macro interface for authors of citation styles. Biblatex's own data backend processor, biber, gives biblatex several advantages, such as dealing with the full range of UTF-8 (Unicode 8.0), which is sometimes necessary when letters and symbols outside the ASCII library are used in a referenced document. The essential settings to compile the reference list are defined in the preamble (file *POHXXXX-preamble.tex*, lines 45 to 47). The command `\printbibliography` in the main document (file *POHXXXX.tex*, line 185) finally prints the list of references.

References are formatted using the numeric style based on the design of the New England Journal of Medicine (NEJM). To choose a different style, the “style” option of the biblatex package would have to be changed in the preamble (file *POHXXXX-preamble.tex*, line 45).

It is recommended to check the information, warnings, and errors issued by biber in the log file (**.blg*), which is created each time the report is compiled.

Tables

The R functions *dataoverview.R*, *paramsum.R*, *datalist.R*, and *printnmr7.R*, supplied in Supplementary Material S2, export data frames straight from R as TeX source code containing the complete floating environment (`\begin{table} ... \end{table}`). These can easily be included via `\input` command (see *results.tex*). This makes most use of the traceability available from subanalyses and data summaries conducted in R, together with the full flexibility of typesetting and formatting available in LaTeX. In addition, the above functions also generate an additional PDF version of the table for preview outside the report. Function *paramsum.R* depends on function *getnmr7.R* for the generation of an Xpose object out of the run archive **.tar.gz*.

The file *POHXXXX-preamble.tex* contains two macros to re-define the table environment, `\dtable` to insert a table including caption and reference label, and `\comptable` to include the desired table and place a footnote under the table (lines 204–220).

The tabular output of multiple population analyzes can be complex and might disrupt the flow of reading if many tables are shown in the core part of the report. It is therefore recommended to show them as “behind-of-text” tables in file *POHXXXX-behindoftextdisplays.tex*.

For some tables, automatic creation is not feasible with the help of an R script or function. Even when using a different environment, such as R Markdown, some information cannot be automatically derived from input tables and must be manually structured and inserted. In the given example, the table that summarizes key characteristics of the clinical studies included in the analysis, file *tables/data-studyoverviewtable.tex*, had to be filled in by hand. This table automatically spans multiple pages if necessary. A footnote can be placed at any position in the table with the `\tnote{<number>}` command to provide more details to the reader. The text of the footnotes is added at the beginning of the file (lines 10 to 11, “Your footnote text”).

Because the table in file *data-studyoverviewtable.tex* contains the whole table environment it is included with the `\input` command (file *data.tex*, line 6).

If tables were created in R, but not through the supplied R functions from this tutorial, the function *datalist.R* can be used to extract any given data frame into LaTeX format.

Figures

Figures and graphs (image files) may exist in different formats. The file format of the image is automatically recognized by LaTeX (**.png*, **.jpg*, **.pdf*, etc.). For graphics presented in the report, the *plotlist.R* function is used to export ggplot objects to the file system as PDF files. For goodness-of-fit plots, *gofplots.R* needs the generation of an Xpose object out of the run archive **.tar.gz* via *getnmr7.R*. For normalized prediction distribution error (NPDE) plots, the generated statistics of the *gof.test()* function of the npde package¹⁷ are included. Generated figures can be placed efficiently at any position in the report by using the plot macros defined in file *POHXXXX-preamble.tex* (lines 84 to 198).

Similar to the tables section, the graphical output of population analyzes can become complex and may disrupt the flow of reading if many figures were shown in the main part. They can be shown as “behind-of-text” displays in file *POHXXXX-behindoftextdisplays.tex*, the same way “behind-of-text” can be used for tables.

Cross references and hyperlinks

Hypertexts are links throughout the document. They help with and improve navigation efficiency for annotations, related sections, references, appendices, tables, or figures that are not located on the same page.¹⁸ References in the LaTeX template are produced by the `cleveref` package¹⁹ via `\cref{[LABEL]}` command at the position where the respective reference should be placed within the text body. This command automatically recognizes the entity to be referenced (table, figure, section, ...) and prints out “Table” and “Figure” etc. accordingly followed by the number. The command `\cref` also accepts a comma separated list of labels (without space), so that they are appropriately concatenated. Two items are abbreviated as “Tables 1 and 2,” three items as “Tables 1 to 3,” etc.

Cross references require that a marker is attached to an entity, such as section, figure, or table. Marker for cross references should be set in the form of `\label{[LABEL]}` (e.g., `\label{sec:item}` or `\label{tab:table1}`). In order to ensure compliance with standards of electronic submission, hyperlinks to external sources, such as web pages, are not allowed. Therefore, the use of the following commands should be avoided:

```
\url{url}: clickable web page
\href[options]{url}{text}: clickable hyperlink
\hyperref[marker]{text}: clickable reference
```

Hyperlinks in PDF are fully controlled by the `hyperref` package.²⁰ For submission, it is usually required that hyperlinks are implemented by rectangles using thin lines or by blue text as appropriate.¹⁸ Bookmarks are required to open to a specific level and PDFs to open with a certain zoom level. These requirements are already considered in the template. The user can control the settings of the `hyperref` package with the command `\hypersetup` (file *POHXXXX.tex*, lines 28 to 49).

The settings of `hyperref` and `cleveref` packages are not defined in *POHXXXX-preamble.tex*, but in the main source file (immediately before `\begin{document}`) to enable some of the variables defined in *POHXXXX-properties.tex* being transferred to the PDF as file (meta) information.

Appendix

The appendix of a population analysis report comprises of any supportive information as defined in the guidelines.²⁻⁴ This can be the analysis plan, listings of code, NONMEM control streams, or other additional information. The appendix is not a standalone document but is placed after the main body of the report. The command `\appendix` activates the LaTeX environment for the appendix (see file

POHXXXX.tex, line 198), which introduces capital letters for numbering the sections, so that the sections of the appendix remain separate from the main text. The appendix is created along with the report when the PDF is compiled, and pagination is automatically performed throughout the whole document.

The skeleton file for the appendix is *POHXXXX-appendix.tex*. PDF files (individual pages from a PDF file or entire PDF documents) can be imported directly by uncommenting the command `\includepdf` (line 16) and writing the correct path. The “page” option of this command allows to select individual or all pages, the “scale” option allows the user to have full control over the scaling process, and the “pagecommand” option declares LaTeX commands, which are executed on each sheet of paper and is useful to control the page header and footer. For more information consult the package documentation.²¹

Any program code must be displayed properly. The command `\lstinputlisting` of the `listings` package²² imports code of many program languages and enables code highlighting for a particular programming language. Syntax highlighting, viewing key words in boldface font, or italicized comments greatly improves readability. Code formatting with the `listings` package is highly customizable through the `\lstset` command. Instead of R code, any other programming language or plain text as for NONMEM control stream and report files can be chosen.

HOW TO FINALLY COMPILE YOUR REPORT – THE LATEXMK SCRIPT

To create the complete report including all indices, different LaTeX programs usually need to be called in a specific order (e.g., `xelatex` or `biber`). This is not only time-consuming, but the log file also needs to be thoroughly examined, to check whether all cross references have been resolved and to decide whether a subsequent compilation run is necessary. The Perl script *latexmk*²³ simplifies the task of PDF generation (compilation). This way, to create a PDF file from the LaTeX document *POHXXXX.tex*, the command “`latexmk POHXXXX`” only needs to be executed in the shell in the `/report` directory (Figure 2). A configuration file (*.latexmkrc*) controls the script. All compilations needed to resolve all cross references are automatically started. It is still recommended to search for “??” in the compiled PDF file, as this indicates an unresolved cross reference.

Another advantage is that the addition, change, or deletion of a file (e.g., a table or figure) in the file system

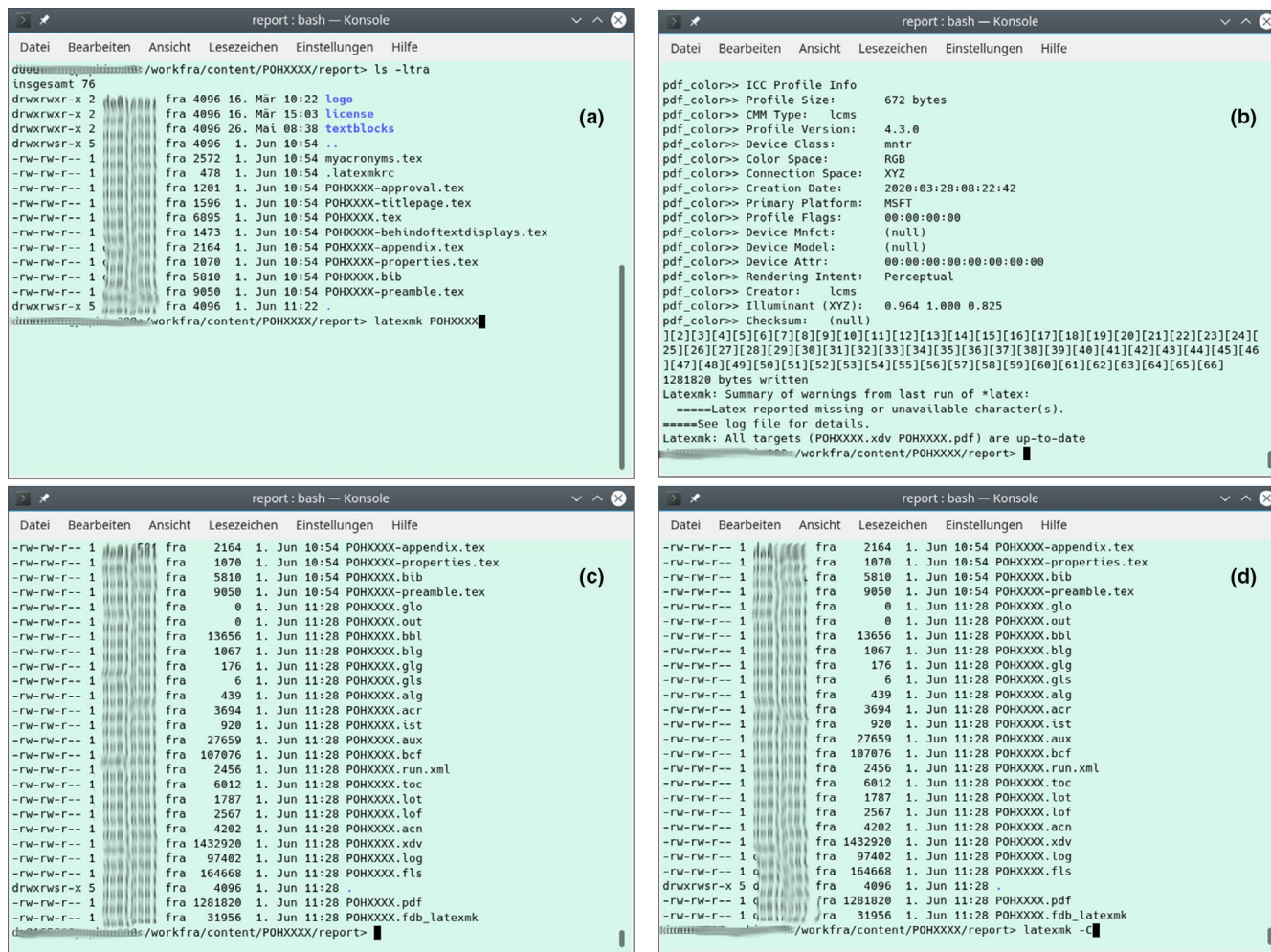


FIGURE 2 Compilation of the complete report through execution of “`latexmk POHXXXX`” command in the command line interface (shell, a), summary message “All targets ... are up-to-date” issued after successful compilation (b), intermediary files created in the file system (c), and cleanup of the intermediary files through execution of “`latexmk -C`” command (d)

is recognized and thereupon it is decided whether a new compilation is necessary or not. By ensuring that a final compilation is triggered shortly before the report is submitted, it is practically impossible for a figure or table to be outdated. The full compilation process ends with a summary statement of whether all cross references were resolved (“All targets ... up-to-date”; Figure 2).

To tidy up the report directory use the command `latexmk -c`. This command removes all automatically generated files, whereas `latexmk -C` also removes the PDF files. Users of LaTeX editor Kile can create an additional button in the toolbar that calls `latexmk`; users of TeXstudio can adapt its configuration accordingly (for detailed description see Supplementary Material S4).

Prior to starting a report, after the template has been created in the directory of choice, a first compilation into a PDF file should be performed and the output checked against the sample PDF file (file `sample.pdf`, Supplementary Materials S2).

DISCUSSION

This tutorial demonstrates how to automate report generation using LaTeX with concrete code examples based on a previous NONMEM tutorial published in this journal.⁸ Ultimately, the approach presented here is intended to contribute to promoting reproducibility of PopPK evaluations within the pharmacometrics community. It is characterized by a low threshold for switching from a word processor to LaTeX, while providing a way to easily adapt the given template to individual needs.

The LaTeX template is supplied together with scripts enabling “housekeeping” of the entire modeling and simulation task that is performed. By creation of automatic directory structures and auxiliary files, every project is started from the same basic structure that automatically feeds into the predefined report architecture, assuring traceability, reproducibility, and making report generation less cumbersome.

One pillar of the automation is the post-processing of NONMEM runs with the R program through additional R scripts supplied with the directory structure. The graphic functions combine the generation and output of the graphic as a PDF file. This way created figures are exported into the respective directory and can automatically be included into the report. Delivered R functions can also create tables and listings as TeX source files, containing the complete LaTeX floating environment. Thus, the standard `\input` command can be used wherever possible to include these objects into the report. These functions also render a complete PDF file of the respective tables/lists outside the report to allow early viewing of the results or use of the results for purposes other than the report.

Reproducibility, which essentially aims at the traceability of the data evaluation process presented in the report, starting with the initial data evaluation and continuing with the programming of the post-processing steps, is becoming increasingly important. Software, such as R Markdown,^{24,25} along with other literate programming options in R or Python, allows the compilation of documents from analysis source files, along with text chunks and to export them to formats like Word, html, and PDF. However, even R Markdown is ultimately just a meta-language built on LaTeX. With its current distribution, R Markdown is less flexible compared to plain LaTeX when specific formatting is needed. Therefore, for now, the use of standard LaTeX is seen as the only option when strict formatting guidelines required by regulatory authorities need to be implemented. The advantage to this tutorial is that with the template and analysis scripts provided, it is easy to perform the analysis and report the results at the same time. In referencing the respective NONMEM runs to unique numbers or standardized names, the corresponding summary tables and goodness-of-fit plots can be automatically linked to the “include” statements within the LaTeX reporting structure. Furthermore, the quality of the report and the underlying analysis is being kept at a high standard as much as possible by reducing between scientist variability.

When comparing with already available solutions based on LaTeX (PharmTeX⁵) parallels can be seen. However, the presented LaTeX template differs in certain aspects from PharmTeX. PharmTeX is a community-based effort, consisting of sophisticated and quite complex LaTeX class, package, initialization files, a document template, platform-independent Perl execution scripts, and portable software bundles (optional). In contrast, the lean template presented here presents a lower entry threshold for users new to the LaTeX environment, enabling them to not only use the given structure but also easily adapt it to their individual needs. Lammarsch and Schubert¹¹ summarize on only three pages the various possibilities to maintain

the writing of reports by means of the established way of LaTeX coding. Therefore, it is consistent that the appendix, cross references, tables, and figure sections are all based on standard LaTeX features. This is accomplished by using KOMA-Script’s “scrartcl” as a document class, which belongs to the standard equipment of LaTeX. KOMA-Script has been actively maintained for many years and comes with excellent user documentation.^{11,12} This also gives the presented template the benefit that it relies purely on stable and semi-standard packages supported externally. Therefore, maintenance needs are reduced to a minimum.

New additional and actively maintained packages are integrated into the KOMA-Script document class ensuring continuity of functionalities. For example, using the combination of xelatex and add-on package fontsetup²⁶ offers the possibility to easily customize current fonts and the matching mathematical fonts. Biber¹⁶ in combination with biblatex is able to deal with many more entry and field types in the *.bib file, UTF-8 encoded.bib files, and provides better sorting control than the classical bibtex program. Many citation styles for biblatex are meanwhile available, which includes the commonly used numeric style based on the design of the NEJM.²⁷ The latexmk package²³ automates the process of creating LaTeX documents by automatically determining and executing the (re)compilation steps. That means, that *latexmk* will repeat the run of latex (or a related program, such as for biber [references], and/or makeglossaries [list of abbreviations]) as often as required to resolve all cross references. It also regularly monitors all the source files to see if any have changed. Every time a change is detected, *latexmk* runs all the programs necessary to generate a new version of the document. This is very practical from the user’s point of view, as the report always reflects the latest status of the analysis.

Although KOMA-Script document classes are compatible with a wide range of add-on packages, rigorous change control must still be applied to ensure changes introduced in new versions do not break things unexpectedly.

The implementation of the LaTeX template presented here is designed to interfere as little as possible with the LaTeX installation. Therefore, all settings-related definitions or meta information are also stored in separate files (e.g., preamble and properties) and are accessible at any time if needed.

Reviewing the report based on the LaTeX source code is difficult, although there are solutions for this as well. One way is to mimic the popular track changes tool available in most word processors.²⁸ However, this itself requires knowledge of LaTeX. In complex organizations, the review of the report often has to be carried out across several hierarchical levels, which means that there is no uniformity in terms of knowledge and skills in the use of

LaTeX. Therefore, the same strategy as with PharmTeX can be applied, where the entire PDF output or individual chapters from LaTeX could be converted into a Word format file by using of a licensed software (e.g., Adobe Acrobat). Additionally, the comment tool feature of any PDF viewer can be used successfully and has been proven to be a good and practicable alternative solution.

The LaTeX template was tested on one Linux distribution (openSUSE 15.1 and 15.3) only, which is related to the fact that the programs used in the tutorial are integrated in a computer cluster. However, the LaTeX template itself runs under other operating systems and with other LaTeX distributions (e.g., MiKTeX), as reported by one of the reviewers of the tutorial. This also applies to the Perl script, in which system-specific modifications (e.g., for *rsync* command) must be included if its usage under Windows (without WSL) is desired. Some of the R functions were developed in the unix domain and might not be executable in the Windows environment without adaptations. This can be achieved with the source code given in the Supplementary Materials S2 and the package dependencies shown in Appendix C.2 of the sample report.

Last but not least, in the authors' experience, using LaTeX for writing population analysis reports has proven successful for more than 20 years by now, and the possibilities the package offers in terms of automation and standardization have paid off many times over the years.

CONCLUSIONS

The automation and LaTeX template of a population analysis report presented in this tutorial should serve as a motivation to start working with LaTeX in a reproducible environment. The embedding of LaTeX in the evaluation process presented here ensures transparency and traceability of the population kinetic analysis along the standard workflow "data, analysis, report." The LaTeX template has got useful default options to meet the requirements of the Electronic Common Technical Document (eCTD) specifications. Nevertheless, in order to adapt the report layout to one's own needs, easy customization without interfering with the LaTeX installation is possible at any time by making the interfaces to the additional packages freely accessible to the user rather than hidden in separate class or style files. Common add-on packages are used that are actively maintained and developed, making them a sustainable choice. The template is versatilely expandable through a multitude of add-on packages and provides latexmk support for easy creation of the complete PDF file, including automatic processing of list of abbreviations and references. An essential part of automation and standardization is the bundling of R functions to provide figures in standardized

file format, image size, and tables as appropriate LaTeX floating environments. This concept allows an extensive use of the corresponding standard functionalities of LaTeX for the inclusion of figures and tables. By integrating suitable scripts, the provision of templates can be automated to avoid time-consuming steps, such as copying and pasting or renaming files. Population analysis reports that were written with the presented LaTeX template were accepted in dossier submissions to regulatory agencies globally.

ACKNOWLEDGEMENTS

The authors would like to thank Robert Priestley. We also thank the anonymous reviewers for helpful comments.

CONFLICT OF INTEREST

All authors are employees from Sanofi company and/or may hold company stock shares.

AUTHOR CONTRIBUTIONS

T.F., S.G., K.L., A.S., A.K. wrote the manuscript; T.F. and S.G. analyzed the data; T.F. and H.S. contributed new reagents/analytical tools.

REFERENCES

1. Dykstra K, Mehrotra N, Tornøe CW, et al. Reporting guidelines for population pharmacokinetic analyses. *J Pharmacokinet Pharmacodyn.* 2015;42:301-314.
2. US Food and Drug Administration (FDA). Population pharmacokinetics guidance for industry. 2019 [cited 03/05/2021] Available from: <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/population-pharmacokinetics>
3. European Medicines Agency (EMA), Committee for Medicinal Products for Human Use. Guideline on reporting the results of population pharmacokinetic analyses. 2007 [cited 03/05/2021] Available from: https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-reporting-results-population-pharmacokinetic-analyses_en.pdf
4. PMDA. Pharmaceuticals and Medical Devices Agency. Guideline on population pharmacokinetic and pharmacodynamic analysis. 2019 [cited 03/05/2021] Available from: https://www.pref.ehime.jp/h25300/tsuuchishuu/seizou/documents/190619_01.pdf
5. Rasmussen CH, Smith MK, Ito K, et al. PharmTeX: a LaTeX-based open-source platform for automated reporting workflow. *AAPS J.* 2018;20:52.
6. Fidler M, Wilkins JJ, Hooijmaijers R, et al. Nonlinear mixed-effects model development and simulation using nlmixr and related R open-source packages. *CPT Pharmacometrics Syst Pharmacol.* 2019;8:621-633.
7. Mittelbach F, Wright J, Schöpf R, et al. The LaTeX Project. [cited 03/05/2021] Available from: <https://www.latex-project.org>
8. Bauer RJ. NONMEM Tutorial Part I: Description of commands and options, with simple examples of population analysis. *CPT Pharmacometrics Syst Pharmacol.* 2019;8:525-537.
9. Speth H. A Linux cluster for population pharmacokinetic analyses. *Int J Clin Pharmacol Ther.* 2004;42:189-190.

10. Wong YJ. Job Scheduling with Maui and Torque. 2011 [cited 03/05/2021] Available from: https://biostat.wustl.edu/sysadmin/UTB_1_MauiTorque.pdf
11. Lammarsch M & Schubert E. latex-refsheet – LATEX Reference Sheet for a thesis with KOMA-Script. 2017 [cited 03/05/2021] Available from: <https://www.ctan.org/pkg/latex-refsheet>
12. Kohm M. Koma-script - A bundle of versatile classes and packages. 2021 [cited 03/05/2021] Available from: <https://www.ctan.org/pkg/koma-script>
13. Talbot N. glossaries – Create glossaries and lists of acronyms. 2021 [cited 03/05/2021] Available from: <https://www.ctan.org/pkg/glossaries>
14. Wikibook LaTeX/Glossary. 2021 [cited 15/05/2021] Available from: <https://en.wikibooks.org/wiki/LaTeX/Glossary>
15. Muller A. TeXMed - a BibTeX interface for PubMed. [cited 03/05/2021] Available from: <http://www.bioinformatics.org/txmed/>
16. Kime P. Biber: a BibTeX replacement for users of BibLaTeX. 2021 [cited 03/05/2021] Available from: <http://biblatex-biber.sourceforge.net/>
17. Comets E, Brendel K, Mentré F. Computing normalised prediction distribution errors to evaluate nonlinear mixed-effect models: the npde add-on package for R. *Comput Methods Programs Biomed.* 2008;90:154-166.
18. ICH M2 Expert Working Group. Electronic Common Technical Document Specification v3.2.2. 2008 [cited 03/05/2021] Available from: https://admin.ich.org/sites/default/files/inlin-e-files/eCTD_Specification_v3_2_2_0.pdf
19. Cubitt T. cleveref – Intelligent cross-referencing. [cited 07/08/2021] Available from: <https://ctan.org/pkg/cleveref>
20. Mittelbach F, Wright J, Schöpf R, et al. hyperref – Extensive support for hypertext in LATEX. [cited 07/08/2021] Available from: <https://ctan.org/pkg/hyperref>
21. Matthias A. pdfpages – Include PDF documents in LATEX. [cited 07/08/2021] Available from: <https://www.ctan.org/pkg/pdfpages>
22. Hoffmann J. listings – Typeset source code listings using LATEX. [cited 07/08/2021] Available from: <https://ctan.org/pkg/listings?lang=en>
23. Collins J, McLean E & Musliner DJ. latexmk – Fully automated LATEX document generation. 2021 [cited 03/05/2021] Available from: <https://www.ctan.org/pkg/latexmk>
24. Xie Y, Allaire JJ, Grolemond G. *R Markdown: The Definitive Guide*. Chapman and Hall/CRC; 2018.
25. Xie Y, Dervieux C, Riederer E. *R Markdown Cookbook*. Chapman and Hall/CRC; 2020.
26. Tzolomitis A. fontsetup – a front-end to fontspec, for selected fonts with math support. 2020 [cited 03/05/2021] Available from: <https://www.ctan.org/pkg/fontsetup>
27. Daniel M. biblatex-nejm – BibLATEX style for the New England Journal of Medicine (NEJM). 2020 [cited 03/05/2021] Available from: <https://ctan.org/pkg/biblatex-nejm?lang=en>
28. Tilmann FJ. latexdiff – Determine and mark up significant differences between LATEX files. 2020 [cited 06/05/2021] Available from: <https://www.ctan.org/pkg/latexdiff>

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Frank T, Gastine S, Lindauer K, Speth H, Strougo A, Kovar A. LaTeX tutorial for the standardization and automation of population analysis reports. *CPT Pharmacometrics Syst Pharmacol.* 2021;10:1310–1322. <https://doi.org/10.1002/psp4.12705>