

Validation Strategies for Report Objects and Programs - Challenges and Obstacles in a Risk Based Reporting World

Dipl. Dok. Christoph Ziegler, F. Hoffmann – La Roche AG, Basel, Switzerland

Agenda I

- Introduction
- Validation Strategies for Report Objects and Programs
 - Validation Process
 - Review of Report Object
 - Standard Code Review
 - Extended Code Review
 - Double Coding
 - Definition
 - Reporting Environment and Access to 1st Line Code and Results
 - Double Coding and ... Specifications / Unclean Data / Source Code Review
 - Automated Comparison of Report Objects
 - Program Promotion on behalf of somebody
 - Modular Analysis Dataset
 - Proc Report Datasets

Agenda II

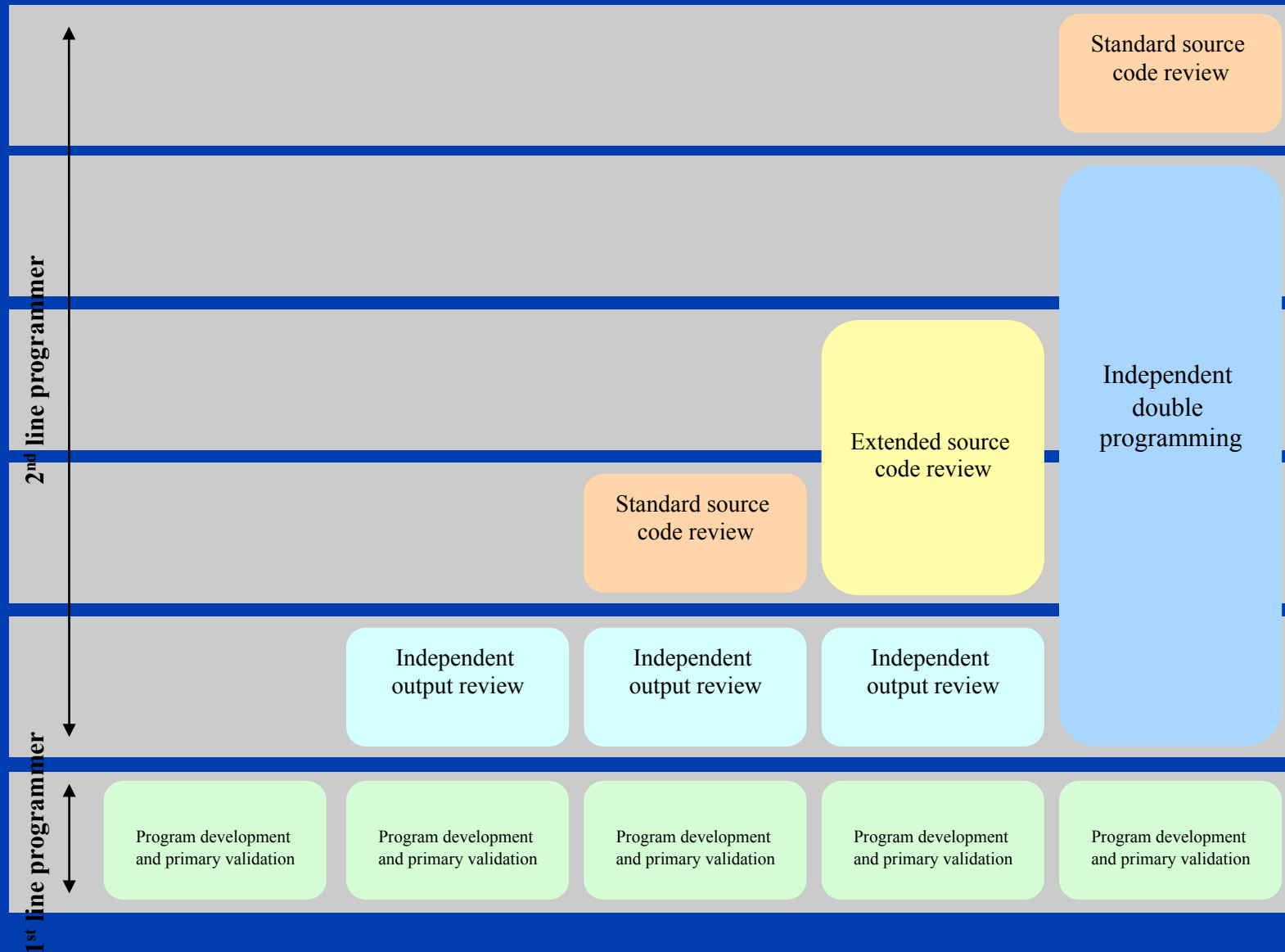
- Other Considerations and Thoughts
 - Change of Report Object Risk – How to control this?
 - How to measure QC-Methods?
 - How to measure Quality of Code?
 - Automated Check of Metadata
 - The Role of the 1st and 2nd Line Programmer
 - Self-Validation?
 - Validation of Report Objects consisting of multiple Programs
 - Data, Programs and Outputs – A Trinity to validate
 - Validation of Specifications
 - Evidence of Validation – Not documented = not done?
 - Is Validation by a QC programmer always needed?
 - Final Execution
 - Training through Validation?
- Conclusion

Introduction I



- Completeness, accuracy and traceability of clinical trial results and summaries are essential with respect to human health and the associated regulatory requirements
- Therefore validation of reporting programs and report objects became common practice but remain a hot topic
- This presentation focuses on validation strategies involving a 2nd line-, i.e. QC-programmer:
 - **review of report objects**
 - **standard code review**
 - **extended code review**
 - **double coding / double programming**
- (Regression and Unit Testing are mainly applied to global standard reporting macros and are not covered)

Introduction II



Validation Strategies for Report Objects and Programs

Validation Process

- In accordance with FDA recommendations the validation method applied to reporting programs and report objects is usually determined based on a risk assessment at the planning stage of a reporting event.
- The respective processes are specified in the company specific SOPs, usually along with associated supportive documents, e.g. checklists and documentation templates.
- A) 1st Line Programmer:
develop → check → review → run
- B) 2nd Line Programmer:
validate → (update by 1st line programmer) → promote to production

Validation Strategies for Report Objects and Programs

Roche

ACCOVION

Review of Report Object

- Regardless of the validation model assigned, review of the report object (minimal method of validation) itself is essential and should at least be done independently.
- Output review comprises but is not limited to the following:
 - Check adherence to requirements and specifications
 - Check accuracy of layout
 - Verify plausibility of results, e.g. via spot checks, checks across outputs, checks against source data
- All report objects should undergo a review cycle before delivery.
- Outputs already validated but recreated at a later stage should be compared against the validated version preferably supported by a programmatic approach, i.e. automatic archival of previous versions and electronic comparison

Validation Strategies for Report Objects and Programs

Roche

Standard Code Review

ACCOVION

- Standard Code Review is characterized as reading through the program source code and checking
 - adherence to underlying specifications
 - logic of the program / plausibility of assumptions made
 - correct selection of datasets / variables / formats
 - correct merging of datasets / handling of exceptions
 - adherence to standards, e.g. program header, comments, GPP
- considered to be the most weak form of program validation (used for low risk report objects). This is because
 - the qc-programmer does not run the source code interactively
 - it is necessary to understand the code and its functionality just by reading through the code
 - certain aspects of the code might be overseen or missed

Validation Strategies for Report Objects and Programs

Roche

Extended Code Review



- Defined as a combination of Standard Code Review and testing of code by running it (or pieces and parts) interactively.
- Code also tested and challenged by running certain parts of the code. This enables the qc-programmer to
 - test and check certain parts of the code or the entire program
 - look at interim datasets / verify the expected results of interim datasets
 - review the log-file thoroughly
 - check that all possible data scenarios are covered by the code
 - understand the code and its functionality better than when just reading through the code
- Disadvantage: code usually does not get re-checked and re-tested once updated data comes in. i.e. qc-programmer needs to check if all possible data scenarios are correctly covered by the program.

Validation Strategies for Report Objects and Programs

Double Coding

Roche

ACCOVION

- Particularly relevant for the validation of high risk report objects.
- Verification of a report object by another program written by the 2nd line programmer and reproducing the results of the report object independently (ADS vs Summary)
- Double programming by an independent partner (process described in paper)
- Double programming in-house (access to source code / specific working environment)
- Physically independent double coding has the advantage that 1st and 2nd line programmer can neither refer to the other's source code nor to output of the program, i.e. both codes are purely based on the specifications.

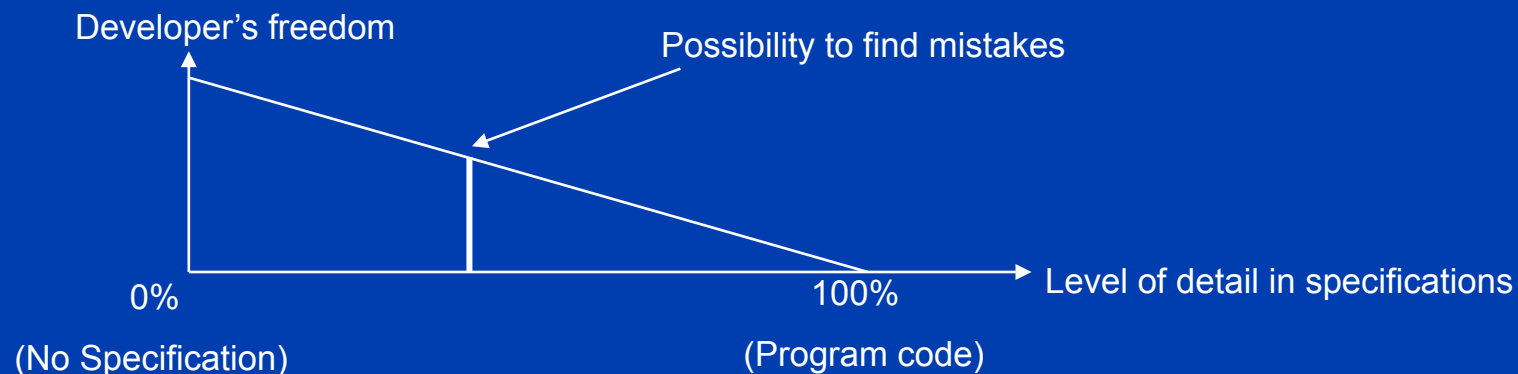
Validation Strategies for Report Objects and Programs

Roche

Double Coding and Specifications

ACCOVION

- Double coding is usually quite resource- and time-consuming and it mainly relies on the available specifications.
- Necessity of high quality and stable specifications.
- On what specifications should double coding be based on?
 - Technical specifications written by the programmer?
 - pure statistical analysis plan written by the statistician?



Validation Strategies for Report Objects and Programs

Roche

Double Coding and Unclean Data

ACCOVION

- Double coding usually needs to take place long before final and clean data is available.
- In many cases time spent for the development of programs is not only dependent on the complexity of the report object but also on the quality and stability of data.
- Specifications are mainly based on the assumption that data are clean and follow predefined data structures and data ranges.
- This is often not the case, especially for interim or DMC analyses where results rely on draft data and changing data.
 - a lot of time is spent to analyze the data and adopt programs accordingly.
- In these cases it needs to be considered if double programming is the adequate method to be applied in terms of validation.

Validation Strategies for Report Objects and Programs

Double Coding and Source Code Review

- In addition to double coding itself the 1st line code should also be checked in terms of programming style, type of code and coding standards in general.
- Programs need to be changed from time to time and the 1st and the 2nd line code should follow the company specific coding standards in order to facilitate maintenance of the code.
- Therefore it is important that the 2nd line programmer also checks the source code (code review) of the 1st line program.
- It is recommended to do that only after successful validation.
- Besides maintenance issues it has to be kept in mind that programs might need to be delivered, e.g. from CRO to sponsor or from sponsor to the authorities.

Validation Strategies for Report Objects and Programs

Automated Comparison of Report Objects ACCOVION

- After successful validation and reconciliation of a report object by double coding, an automated comparison should take place.
- I.e. the 2nd line program should always be run when new or updated data comes in order to detect possible problems or discrepancies between the 1st and 2nd line code due to new data constellations or unforeseen new data.
- Ideally the comparison of the report objects should be done at the end of the 2nd line program using PROC COMPARE.
- It is of advantage to name the 2nd line program as the 1st line program with a suffix (e.g. demo.sas and demo_qc.sas) in order to have both programs next to each other in the company specific version control system.

Validation Strategies for Report Objects and Programs

Program Promotion on Behalf of Somebody



- When working in different areas where the 2nd line programmer does not have access to the 1st line programmer's code, program promotion "on behalf of" is necessary.
- This means: In a program version control system, a 3rd person has to promote the 1st line code on behalf of the 2nd line programmer after successful validation and reconciliation because he or she does not have access to the 1st line code.
- This needs to be documented with a reference to the validation and reconciliation documentation or log files.

Validation Strategies for Report Objects and Programs

Roche

Modular Analysis Datasets

ACCOVION

- Another aspect of double coding is that specifications (technical and statistical analysis plan) sometimes do not come in all at once and that specifications change from time to time.
- Therefore a modular analysis dataset structure is recommended, i.e. each variable of an analysis dataset should ideally have its own program or macro.
- This makes it easier to maintain the 1st and 2nd line code and avoids long “spaghetti” code.
- When adding a new variable or when changing an existing variable it is ensured that only the affected variable gets touched and that other variables remain untouched.
- (see paper for an example)

Validation Strategies for Report Objects and Programs

PROC REPORT Datasets

- The decision on what to double code should be based on risk assessment at the planning stage of a reporting event.
- Analysis datasets are usually of high risk because the included database and derived variables are often used for more than one summary or listing report object.
- Analysis datasets are the basis of reporting and therefore of high importance.
- If summary report objects get double coded, it is recommended that only the content (the numbers) get reproduced, not the entire layout.
- This is because reproducing the entire layout is resource intensive and has no added value.
- It is recommended to use so-called PROC REPORT datasets with a pre-defined structure
- A PROC REPORT dataset is defined as the dataset going into the SAS procedure PROC REPORT in order to display results (e.g. summary statistics) and other variable content.

Other Considerations and Thoughts

Change of Report Object Risk – How to control this?

- During the conduct of a project, the risk of a report object or a reporting program can change. E.g. a summary report object which had a low risk initially and where standard code review was performed can suddenly become of high risk due to safety issues in a clinical trial. Double coding may now be required. Questions related to the process arise:
 - Is there a process in place triggering the review and revision of the risk associated with the report object?
 - Is the company specific reporting environment and program version control system able to track changes like this?
- Review of the risk assessments at particular time-points should be included in the company specific standard operating procedure and should at least take place before delivery of the final results.
- It has to be taken into account that higher risk is associated with additional effort, especially as report objects are usually produced using multiple other programs and macros.

Other Considerations and Thoughts

How to measure QC-Methods?

- Which validation method is the best and most effective one?
- In general, it is quite difficult to measure the effectiveness and the success of the different qc-methods for report objects and programs.
- In order to learn for the future and in order to see whether the correct qc-methods were applied, a system or a process needs to be in place which checks the amount of errors and which compares this to the applied qc-method.
- The problem with this approach is that it is not reliable.
- Double coding (and the other validation methods as well) mainly depends on the quality of the specifications.
- The success and the quality of Code Review and Extended Code Review are highly dependent on the experience and the level of the qc-programmer.
- QC methods can finally only be measured by the number and importance of errors identified in final results.
- Issues in report events are usually determined by chance only and therefore no reliable assessment of QC methods is available.

Other Considerations and Thoughts

How to measure the Quality of Code?

- When performing Code Review or Extended Code Review the question is how to measure the quality of the code which is reviewed.
- Quality of code could be judged by
 - the amount of comments in the code?
 - the amount of feedback loops with the 1st line programmer?
 - the correct results? the programming style?
 - the ability of the code to handle different data scenarios?
 - the time spent for maintenance by another programmer?
- It is very difficult to measure the quality of code.
- Sophisticated instruments would be needed to perform this task. However, this would be quite time consuming and resource intensive.
- Another question which comes when performing code review is when is it time to say that a program is ok. In most of the cases this is purely based on the gut feeling of the 2nd line programmer.
- An experienced qc-programmer might confirm that a program works as expected earlier than an inexperienced qc-programmer who is performing the code review several times in order to be sure that all is fine.

Other Considerations and Thoughts

Automated Check of Metadata

- Besides checking the programming results, the 1st and 2nd line programmer should both verify that the metadata also complies with the respective specifications.
- This can be quite time-consuming and error-prone if length, format, label for dozens of variables need to be checked manually.
- The implementation of an automated check procedure including consistency checks across panels can facilitate programming as well as the validation process.

Other Considerations and Thoughts

The Role of the 1st and 2nd Line Programmer

- Though having validation methods in place, the main responsibility to ensure correctness and quality should rest with the 1st line programmer.
- The role of the 2nd line programmer is to provide added insurance of quality to programs and report objects, regardless of the validation method which is applied.
- This means, the 1st line programmer should never rely on the 2nd line programmer.
- It is in the nature of statistical 1st line programmers to produce high quality programs and report objects, i.e. not to rely on the 2nd line programmer.

Other Considerations and Thoughts

Self-Validation?

- One problem when validating reporting programs in a version control-system is so-called “self-validation”.
- This situation might happen, when a program gets promoted by programmer A from the development area to the qc area.
- Then programmer B should ideally validate the code and promote the code to production.
- However, when programmer B does e.g. a further small update of the code he / she checks the program back out to development, updates the code and brings it back to the validation area.
- If then programmer A promotes the program code to production level, it is so-called self-validation which is not acceptable.

Other Considerations and Thoughts

Validation of Report Objects consisting of multiple Programs

- Quite often, report objects are produced using multiple (standard) macros and other programs.
- When double coding a report object which is created using multiple programs, it is recommended to create a stand-alone 2nd line program which is not using other macros and programs.
- There are two reasons why this is of advantage:
 - all programs and macros which are used in the 1st line program to create the report object get (partly) validated as well
 - when submitting programs to the FDA, stand-alone programs are requested. The 2nd line code can be used for this purpose once validation is completed successfully.

Other Considerations and Thoughts

Data, Programs and Outputs – A Trinity to validate

- When validating report objects by Extended Code Review or Double Coding, it is important to always look at the data, the program and the actual output when performing the validation (besides the underlying specifications).
- Usually 1st line programs are developed on non-final data during the course of a clinical trial for a certain reporting event.
- This is also when validation takes place.
- It is important to re-check report objects on the final data for a reporting event again.
- In case Double Coding is applied, the 2nd line program should be re-run and the comparison should take place on the final data.
- In case Extended Code Review takes place, re-checking of the code is recommended as well.

Other Considerations and Thoughts

Validation of Specifications

- Programs and the resulting report objects mainly rely on different kinds of specifications, e.g. Statistical Analysis Plan, dataset specifications, table shells, etc.
- Currently specifications are challenged during the programming and validation period, but formal validation of specifications is still no common practice.
- During the process of double coding issues in the specifications are usually identified and corrected accordingly.
- But with the approach of source code review this is often not the case.
- Therefore it should be considered whether review and approval of specifications is necessary.
- At least when providing specifications to a third party, e.g. a CRO providing specifications to the sponsor for review and acceptance testing, the implementation of an internal review cycle is strongly recommended.

Other Considerations and Thoughts

Evidence of Validation – Not documented = not done?

- Validation activities need to be documented in order to have a proof that validation was performed.
- Generally the following documentation is recommended and should be completed continuously
 - Tracking of validation steps performed including dates and responsibilities
 - Validation findings and actions taken
 - Final approval
- This information can be collected either manually or within a version tracking system.
- It has to be kept in mind, that validation documentation needs to be available for internal or external audits and might be required by sponsors for archival.
- The documentation process and the associated document templates are usually described in the company specific SOPs together with checklists that need to be followed during the validation process.

Other Considerations and Thoughts

Evidence of Validation – Not documented = not done?

In addition to the documentation mentioned above different forms of evidence are required by different validation methods.

- **Double Coding:**

- It is recommended to make a reference to the 2nd line qc log and output files when promoting the 1st line program to production within the company specific version control system.
- Those log file and output files should permanently be stored and should not be overwritten. Also a reference to the 2nd line code is strongly recommended.

- **Extended /Standard Code Review:**

- It is difficult and time-consuming to document what is done during an Extended or Standard Code Review.
- It is recommended to set up a standard check-/ test-list
- Additional tests and checks can be documented if needed.
- By promoting program code to production in the version control system, the qc-programmer confirms that the standard and pre-defined check- / test-list were followed.

Other Considerations and Thoughts

Is Validation by a QC-Programmer always needed?

- One common standard approach within the statistical programming departments in the pharmaceutical industry is to apply Standard Code Review for low risk report objects, Extended Code Review for medium report objects and Double Coding for high risk report objects.
- As mentioned in the chapter “the role of the 1st and 2nd line programmer”, the main responsibility for ensuring correctness and quality of a program should rest with the 1st line programmer.
- For low risk report objects it is sometimes acceptable to skip validation by a qc programmer.
- This saves time and resources and puts more responsibility on the 1st line programmer.

Other Considerations and Thoughts

Final Execution

- Validation usually takes place before final data is available and validation activities can neither be postponed until then, nor repeated at that point in time.
- Therefore a formal process should be in place for the final execution of the programs.
- For final execution all programs need to be final, i.e. production. For this final run it needs at least to be checked, that
 - Report objects are generated from correct version of data
 - SAS logs document proper run of the programs
 - All outputs are complete and created from the final program run
 - For programs where a qc program is in place, the results of the respective programs are compared against the results of the initial programs
 - For outputs where a validated version on the same data status is

Other Considerations and Thoughts

Training through Validation?

- The aspect of training associated with any kind of code review should not be underestimated.
- By reviewing other programmers' code, even experienced programmers identify coding solutions they were not aware of.
- On the one hand code review should be done by experienced programmers in order to identify possible problems within the code on the other hand it is a good opportunity for younger colleagues to expand their knowledge.
- If less experienced SAS-programmers do code review for validation purposes the code review should always be accompanied by other validation means, e.g. partial double programming.

Conclusion

- Validation is crucial with respect to quality of report objects.
- Therefore validation is one of the most important tasks within each programming department.
- Efficient and effective validation procedures are required to meet quality expectations while adhering to time and resource constraints at the same time.
- Due to the high complexity of the analyses being performed and the huge amount of data to be handled error-free reporting events can hardly be achieved, even though processes and standards are constantly improving.
- There is no validation process in place ensuring 100% correctness of a reporting event.
- The challenge of identifying the ideal validation method for each single report object remains.
- Writing of programming specifications, generating of reporting programs as well as quality control mainly relies on the know-how and experience of the programmers involved.

Thank You



Contact the authors at:

Christoph Ziegler, Dipl.-Dok, Senior Lead Statistical Programmer, QC Process Site SME Basel

F. Hoffmann – La Roche AG

Malzgasse 30, Bldg. 670

4070 Basel

+41 61 68 89200

christoph.ziegler@roche.com

www.roche.com

Beate Hientzsch, Director, Statistical Programming

Accovion GmbH

Helfmann-Park 10

65760 Eschborn, Germany

+49-6196-7709-406

beate.hientzsch@accovion.com

www.accovion.com

Brand and product names are trademarks of the respective companies.