



# Use of gatekeeping strategies in complex study designs

Delphine Anthony

## ► To cite this version:

Delphine Anthony. Use of gatekeeping strategies in complex study designs. Methodology [stat.ME]. 2010. dumas-00517484

**HAL Id: dumas-00517484**

**<https://dumas.ccsd.cnrs.fr/dumas-00517484>**

Submitted on 14 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



*UFR de mathématique  
et d'informatique  
Université de Strasbourg,  
France*



*Global Clinical R&D Department  
GlaxoSmithKline Biologicals  
Wavre,  
Belgium*

# **Use of Gatekeeping Strategies** **in complex study designs**

**Feb. – Aug. 2010**

**Delphine Anthony**

**Master 2 of Sciences, Technologies, Health,  
Mention Mathematics and applications,  
Specialty Statistic**

**Internship Tutor: Dorothée Meric**

## ABSTRACT

In clinical trials, problems arise frequently when multiple tests are performed. Multiplicity may have a substantial influence on the rate of false positive conclusions. Therefore, control of Type I error rate  $\alpha$  is an important concern when designing clinical trials. Several statistical approaches have been developed to allow this control in case of multiple objectives.

This work focuses on the use of gatekeeping strategies for multiplicity adjustments in case of multiple objectives in clinical trials.

The principle of gatekeeping strategies is to test in a sequential manner objectives which are hierarchically ordered. Depending on the study design, different types of gatekeeping strategy can be used to control the Type I error: serial, parallel and tree-structured gatekeeping strategies.

Here, an illustration of two strategies is exposed. We evaluate the performance of three different scenarios, two serial scenarios and a parallel one, with multiple objectives through several simulations of clinical trial data. We also study the performance variations in case of correlated data. We find that increasing the correlation may have an effect on the performance of scenarios.

# Table of Contents

Thanks.....	5
Table of Abbreviations .....	6
Introduction .....	7
I. Pharmaceutical Context .....	8
I.1. Presentation of the Company.....	8
I.2. Presentation of the Department .....	8
I.2.1. Composition .....	8
I.2.2. Organisation by Team .....	8
I.3. Role and Responsibilities of a Biostatistician .....	9
I.3.1. Practical Work.....	9
I.3.2. Research Work.....	9
II. Description of the Problem.....	10
II. 1. Overview of Multiplicity Problems in Clinical Trials .....	10
II. 2. Hypothesis Testing.....	11
II. 2. 1. Definitions .....	11
II. 2. 2. Alpha Adjustment.....	12
III. Description of the Methods .....	14
III. 1. Multiple Testing Procedures.....	14
III. 1. 1. Bonferroni Procedure .....	14
III. 1. 2. Holm Procedure .....	14
III. 2. Gatekeeping Strategies .....	15
III. 2. 1. Serial Strategy .....	16
III. 2. 2. Parallel Strategy .....	17
III. 2. 3. Tree-structured Strategy.....	19
IV. Simulations.....	22
IV. 1. Settings .....	22

IV. 1. 1. Study Explanation .....	22
IV. 1. 2. Objectives .....	24
IV. 1. 3. Hypotheses Tested .....	24
IV. 2. Gatekeeping Scenarios.....	25
IV. 3. Simulated Data .....	27
IV. 3. 1. Normal Distribution.....	27
IV. 3. 2. Correlation Cases .....	29
IV. 4. Performance Computing .....	30
V. Results and Analysis.....	32
V. 1. Tables .....	32
V. 2. Analysis.....	33
VI. Conclusions and Discussion .....	35
VII. References.....	36
VIII. Appendixes.....	37
Appendix A: SAS Macro for simulations of datasets for scenarios 1 and 2 .....	38
Appendix B: SAS Macro for testing scenario 1 (serial) .....	42
Appendix C: SAS Macro for testing scenario 2 (serial) .....	45
Appendix D: SAS Macro for simulations of datasets for scenario 3 .....	50
Appendix E: SAS Macro for testing objectives of scenario 3 .....	54
Appendix F: SAS Macro for computing performance of scenario 3 (parallel).....	55

## Thanks

First, I would like to thank Marc Fourneau, the Biometrics Director who accepted me in his team for six months.

I thank in particular Dorothée Meric, my tutor, for her support during all my internship. She helped me to understand pharmaceutical issues through her explanations and was very implicated in my work.

Many thanks to Mohamed El Idrissi, Yang Feng and Fabian Tibaldi, members of the “multiplicity group”. Their multiple advices were useful and helped me to orientate my researches.

I thank a lot Gregory Catteau, Marie-Pierre David and Toufik Zahaf, HPV statisticians, for their cheerfulness as well as for sharing their knowledge and experience.

A big thank to Marie Lebacq for her availability and for having reviewed many times my document.

Thanks to Michael Povey who were always there to answer to my questions, even in the shuttle.

Special thanks to Mitchell who often made me laugh. Days at work would not have been the same without him.

Finally, thanks to all the HPV team, all GSK statisticians and other people I met during my internship. Thank you for your welcome and your encouragements.

## Table of Abbreviations

Bonf.	Bonferroni
cf.	Confer
corr.	Correlation
Diff.	Difference
E	Endpoint
F	Family
FWER	Familywise Error Rate
GCRD	Global Clinical Research and Development
GM	Geometric Mean
GMT	Geometric Mean of Titres
GSK	GlaxoSmithKline
H	Hypothesis
HPV	Human Papillomavirus
i.e.	That is
med.	Medium
N. E.	Numerical Example
Non-inf, N-I	Non-inferiority
O	Objective
P	Primary
p-val.	P-value
R&D	Research and Development
S	Secondary
Sim.	Simulation
Super	Superiority
T	Tertiary
vs.	Versus

# Introduction

Let us suppose that you take a test without having studied for it. This test is a double-choice questionnaire and you answer randomly to each question. If the number of questions is important, you may answer correctly to one or more questions just by chance. And so, your score, which should have been zero, will be positive.

This is a simple example of multiplicity problems which occur frequently in clinical trials. Indeed, by performing a lot of tests on a new treatment, a benefit effect of the treatment can be showed just by chance. To avoid this, adjustments are needed, especially in case of complex study designs, i.e. formulations of trials and experiments with multiple objectives. Several methods for these adjustments are valid as the Bonferroni and Holm method. We will briefly describe them and explain principally the use of gatekeeping strategies, a new more efficient method, in complex study designs.

The objective of this work was, firstly, to understand three types of gatekeeping procedures in the context of multiplicity problems and vaccine clinical trials. Then, the second objective was to apply the methods on some practical situations. Finally, the third objective was to compare, using SAS, the performances obtained with various methods of Type I error adjustment (Bonferroni, Holm, gatekeeping strategies).

In a first part, we will be interested in the pharmaceutical context of this work with the presentation of the company, the department and also the role and responsibilities of a statistician.

Then, we will give a description of the problem with an overview of multiplicity problems in clinical trials and precisions about the hypothesis testing.

The third part will describe the methods used for adjustments, as multiple testing procedures given by the Bonferroni and the Holm methods, and the gatekeeping method with the three different strategies.

In the fourth part, an example applied to two gatekeeping strategies will be exposed. Starting from real study data, we will evaluate the performance of three different scenarios, two serial scenarios and a parallel one, through several simulations. We will also study the performance of each scenario in case of correlated data.

In the fifth and last part, we will show the simulations results and analyze them.



# **I. Pharmaceutical Context**

From February to August 2010, I worked as an intern for GlaxoSmithKline Biologicals (GSK Biologicals), in the Global Clinical Research and Development (GCRD) department.

## **I.1. Presentation of the Company**

GSK Biologicals, part of the GlaxoSmithKline Group, is one of the world's leading vaccine companies. Headquartered in Belgium, the company is active in the fields of vaccine research, development and production with over 30 vaccines approved for marketing and 20 more in development. GSK Biologicals has one of the strongest pipelines in the industry as it is working to develop or improve vaccines to cover a range of global diseases including HIV, influenza, malaria and tuberculosis.

## **I.2. Presentation of the Department**

### **I.2.1. Composition**

The GCRD department includes Worldwide Clinical R&D Vaccines activities, Clinical Safety and Pharmacovigilance, Epidemiology and Clinical/Medical Compliance activities. In fact, all necessary functions required to face the future challenges in clinical development and to ensure post licensure monitoring of the vaccines have been brought together.

### **I.2.2. Organisation by Team**

The department is divided into several teams corresponding to the different vaccine projects. The main functions composing a clinical project team are Clinical Development Manager, Global Study Manager, Data Manager, Statistician, Scientific Writer, and Regulatory Affairs.

During my placement, I was in the Human Papillomavirus' (HPV) workplace. The HPV causes cervical cancer development in case of persistent infection. Cervical cancer is the second most common cancer among women under 45 years of age.

In the open space, all the functions are grouped together and people are interacting with each other to organise their work, as each step of vaccine development involves different responsibilities. Two senior specialist and three specialist biostatisticians work for this project. And so, I have noticed that statisticians are very involved at all stages present in vaccine development.

### **I.3. Role and Responsibilities of a Biostatistician**

#### **I.3.1. Practical Work**

The research and the development of a new vaccine before launching it on the market is a long procedure which often exceed 10 years. In the pharmaceutical industry, statisticians are needed during pre-clinical and clinical studies, marketing and manufacturing operations.

In the HPV project, clinical studies are in process. Performing several studies is necessary in a project in order to better understand the effects of the new vaccine. For each study, biostatisticians work on the protocol, decide the study design, prepare the randomisation and perform analysis of results. They also help summarizing data to make it understandable for non-statisticians, interpret results and draw statistical conclusions.

#### **I.3.2. Research Work**

As statistic is an evolving science, a biostatistician always needs to be aware of new statistical methods used or experimented in R&D. To respond to that fact, there are forums and meetings organised at GSK every week where a statistician exposes some unknown processes or explains their application to others.

In the same way, some statisticians from different projects can form a group and do researches on a particular topic. In this context, I integrated the “multiplicity project group” in which five GSK statisticians are dealing with multiple testing problems in clinical trials.

During my internship, I often presented them my results and they helped me to interpret them and to go into details.

## II. Description of the Problem

In the following, we will introduce multiplicity problems and the need of adjustment in clinical trials.

### II. 1. Overview of Multiplicity Problems in Clinical Trials

A clinical trial is a research study on human volunteers, called subjects, to answer specific health questions. Several clinical trials are performed in vaccine development. They can be classified into four phases.

The phase I of the medical research assess the safety and tolerability of a vaccine. The phase II clinical trials are designed to determine the optimal dose of new treatments to administer while respecting patient safety. In phase III, trials are conducted to provide evidence that new treatments are safe and effective in treating of targeted diseases. They are also called confirmatory controlled clinical trials, as they take place after the phase II trials.

In phase III, the most serious risk in a clinical view is to achieve false positive conclusions on the effectiveness of a new treatment. Precautions have to be taken to avoid that, especially regarding the multiplicity problems.

This kind of problems depends on how trials are designed. The number of study objectives has influence on multiplicity as the fact that there are different time points and dose levels to compare. In vaccine studies, we talk about multiplicity problems in two situations: in case of multiple comparisons and in case of multiple endpoints<sup>1</sup>.

Multiple comparisons are usually performed in trials with multiple doses compared to a common control or trials with multiple subgroups.

On the other hand, trials with multiple endpoints are generally done when the efficacy of an experimental treatment needs to be assessed on multiple outcome measures. As the most common multiplicity problem is observed in case of multiple endpoints studies, we will develop only this part afterwards.

---

<sup>1</sup> An endpoint refers to occurrence of a disease, symptom, sign or laboratory abnormality that constitutes one of the target outcomes of the trial. One or more objectives can be associated to one endpoint.

## II. 2. Hypothesis Testing

In this section, we introduce definitions and notations about hypothesis testing and give details on the alpha adjustment.

### II. 2. 1. Definitions

#### Superiority and Non-inferiority

A clinical study design includes one or several objectives and a test of hypotheses is generally associated to each objective. Statistical tools can be used to show, among others, superiority or non-inferiority of the new treatment over the comparator treatment.

Let us note  $H_0$  as the null hypothesis,  $H_1$  as the alternative one, A as the new treatment and B as the comparator treatment.

The test of superiority of A compared to B can be written as it follows, in a simplified way.

$$H_0: A \leq B \quad \text{vs.} \quad H_1: A > B$$

*The hypothesis that A is inferior or equivalent to B should be rejected, at a certain significance level, to show the superiority of A compared to B.*

The test of non-inferiority of A compared to B can be written as it follows, in a simplified way.

$$H_0: A < B \quad \text{vs.} \quad H_1: A \geq B$$

*The hypothesis that A is inferior to B should be rejected, at a certain significance level, to show the non-inferiority of A compared to B.*

#### Type I and Familywise Error Rate

In testing a single objective to show efficacy, a statistical test can lead to conclude that the treatment is effective just by chance, whereas the treatment has no effect. This error is known as the Type I error or the false positive error. Its rate is noted  $\alpha$ .

In mathematical terms, the Type I error rate is the probability of rejecting the null hypothesis ( $H_0$ ) when this hypothesis is true.

*Testing the objective O consists in testing:  $H_0$  vs.  $H_1$*

*and the type I error rate is:*

$$\alpha = P(\text{reject } H_0 \mid H_0 \text{ true})$$

By testing several hypotheses, the Type I error rate has to be controlled for each hypothesis as well as the maximum overall Type I error rate. This error rate is also called the Familywise Error Rate (FWER).

The FWER is the probability of rejecting at least one hypothesis when all hypotheses are true.

*Testing the objective  $O_i$  consists in testing:  $H_0^i$  vs.  $H_1^i \quad \forall i = 1, \dots, m$*

*and the FWER for the set of objectives  $\{O_1, \dots, O_m\}$  is:*

$$\alpha_{FWER} = P(\{\text{reject } H_0^1\} \cup \dots \cup \{\text{reject } H_0^m\} \mid \{H_0^1 \text{ true}\} \cap \dots \cap \{H_0^m \text{ true}\})$$

## II. 2. 2. Alpha Adjustment

To introduce the need of adjustment, we will consider an example.

We would like to reach 4 objectives ( $O_1, O_2, O_3$  and  $O_4$ ) by testing 4 null hypotheses ( $H_1, H_2, H_3$  and  $H_4$  respectively) at  $\alpha_1, \alpha_2, \alpha_3$  and  $\alpha_4$  (resp.) significance levels. We assume that each objective is associated with one endpoint and that all endpoints are independent.

$\forall i = 1, \dots, 4 \quad O_i$ : Test of  $H_i$  at  $\alpha_i$  level

- 1<sup>st</sup> case of testing:  $O_1$  AND  $O_2$  AND  $O_3$  AND  $O_4$

Here, we fix us a condition which is to reach all the objectives at the same time otherwise no conclusion would be possible. In other words, we need to reject all the null hypotheses associated with the objectives. If one or more hypotheses failed to be rejected, we would accept all of them.

*Numerical Example:*

$$\forall i = 1, \dots, 4 \quad \text{If } \alpha_i = 0.05 \quad \text{then } \alpha_{FWER} = 0.05$$

The advantage of this situation is that it does not need any alpha adjustment. The inconvenient is that the condition of rejecting all hypotheses at the same time is very difficult to obtain and is not flexible.

- 2<sup>nd</sup> case of testing:  $O_1$  OR  $O_2$  OR  $O_3$  OR  $O_4$

In this case, we are allowed to fail rejecting one or more hypotheses and still have conclusions for the others which are rejected. This kind of scenario is less restrictive and so, it is more used in clinical trials where several objectives are tested.

*N.E.:*

$\forall i = 1, \dots, 4$  If  $\alpha_i = 0.05$  then  $\alpha_{FWER} = 0.1855$  and not 0.05

In fact, the formula to calculate the FWER for this scenario is:

$$\alpha_{FWER} = 1 - (1 - \alpha_1)(1 - \alpha_2)(1 - \alpha_3)(1 - \alpha_4)$$

This formula is obtained as it follows.

$$\begin{aligned}\alpha_{FWER} &= P(\{reject H_0^1\} \cup \dots \cup \{reject H_0^m\} \mid \{H_0^1 \text{ true}\} \cap \dots \cap \{H_0^m \text{ true}\}) \\ &= P(\{reject H_0^1 \mid H_0^1 \text{ true}\} \cup \dots \cup \{reject H_0^m \mid H_0^m \text{ true}\}) \\ \Leftrightarrow 1 - \alpha_{FWER} &= P(\{accept H_0^1 \mid H_0^1 \text{ true}\} \cap \dots \cap \{accept H_0^m \mid H_0^m \text{ true}\}) \\ &= (1 - \alpha_1) \dots (1 - \alpha_m) \\ \Leftrightarrow \alpha_{FWER} &= 1 - (1 - \alpha_1) \dots (1 - \alpha_m)\end{aligned}$$

This example reflects the problem caused by testing multiple objectives without any condition, which is the inflation of the FWER.

The need of adjustment is real and we will see that several methods are available to control the FWER.

### III. Description of the Methods

In the following, we will note the FWER as  $\alpha$  and the adjusted  $\alpha$  corresponding to  $H_i$  as  $\alpha_i$  for each  $i$  from 1 to  $m$ , where  $m$  is the number of null hypotheses tested associated to the objectives in the trial. We introduce  $p_i$  as well, which is the p-value associated to the test of  $H_i$  and so, we are able to reject  $H_i$  if  $p_i$  is smaller than  $\alpha_i$ .

In this section, we will introduce three adjustment methods, the Bonferroni and the Holm methods which are multiple testing procedures and the gatekeeping method with its strategies.

#### III. 1. Multiple Testing Procedures

In this section, we will be interested in two important classes of multiple testing procedures which are single-step and stepwise tests. Single-step methods test each hypothesis of interest independently of the other hypotheses whereas stepwise methods test one hypothesis at a time in a sequential manner. There are several methods which are parts of these two classes. We will describe two of them here: the single-step Bonferroni test and the stepwise Holm test.

##### III. 1. 1. Bonferroni Procedure

Adjustment by Bonferroni method is perhaps the most widely used multiplicity adjustment in clinical trials. It consists to test all the null hypotheses associated with the objectives at a same adjusted alpha level. The order in which the hypotheses are examined is not important.

$$\begin{aligned} & \text{Testing } H_1 \text{ OR } H_2 \text{ OR } \dots \text{ OR } H_m \\ \Leftrightarrow \quad & \forall i = 1, \dots, m \quad \text{Reject } H_i \text{ if } p_i \leq \alpha_i = \alpha/m \end{aligned}$$

*N. E.*

If  $\alpha = 0.05$  and  $m = 4$  then  $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.05/4 = 0.0125$

This method controls the FWER but it is also very conservative. In other words, the adjusted alpha for each hypothesis is considerably smaller than it needs to be to control the FWER.

##### III. 1. 2. Holm Procedure

For this procedure, we need to order the p-values for all the hypotheses tested. We note as  $p_{(i)}$  the  $i^{\text{th}}$  p-value,  $H_{(i)}$  its hypothesis associated and we have  $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(m)}$ . The procedure starts with the hypothesis associated with the most significant p-value,  $H_{(1)}$ .

- Step 1:      Test  $H_{(1)}$ 
  - If  $p_{(1)} \leq \alpha_{(1)} = \alpha/m$  then reject  $H_{(1)}$  and go to the next step.
  - Otherwise, retain all the hypotheses and stop.
  
- Step  $i = 2, \dots, m$ :      Test  $H_{(i)}$ 
  - If  $p_{(i)} \leq \alpha_{(i)} = \alpha/(m - i + 1)$  then reject  $H_{(i)}$  and go to the next step.
  - Otherwise, retain  $H_{(i)}, H_{(i+1)}, \dots, H_{(m)}$  and stop.

N. E.

$\alpha = 0.05$  and  $m = 4$

- If  $p_{(1)} \leq \alpha_{(1)} = 0.05/4 = 0.0125$  then reject  $H_{(1)}$  and test  $H_{(2)}$ .  
Otherwise, retain  $H_{(1)}, H_{(2)}, H_{(3)}, H_{(4)}$  and stop.
- If  $p_{(2)} \leq \alpha_{(2)} = 0.05/3 = 0.01667$  then reject  $H_{(2)}$  and test  $H_{(3)}$ .  
Otherwise, retain  $H_{(2)}, H_{(3)}, H_{(4)}$  and stop.
- If  $p_{(3)} \leq \alpha_{(3)} = 0.05/2 = 0.025$  then reject  $H_{(3)}$  and test  $H_{(4)}$ .  
Otherwise, retain  $H_{(3)}, H_{(4)}$  and stop.
- If  $p_{(4)} \leq \alpha_{(4)} = 0.05$  then reject  $H_{(4)}$ .  
Otherwise, retain  $H_{(4)}$ .

This method controls the FWER but is less conservative than the Bonferroni one and so, it is more powerful. However, p-values need to be ordered and the hypotheses are tested in a sequential manner in opposition to the Bonferroni method. In fact, because of this, the Bonferroni method is usually preferred to the Holm one in clinical trial testing.

### III. 2. Gatekeeping Strategies

Another method to control the FWER is the gatekeeping method, introduced by Alex Dmitrienko et al. (cf. Reference [4]). This method enables to control the risk of false efficacy claims with respect to multiple ordered analyses.

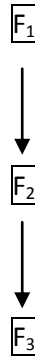
Its principle is to organise the multiple objectives in a hierarchical structure and examine ordered analyses in a sequential manner beginning with the primary objectives. To do this, we define several families of analyses:  $F_1, \dots, F_n$ .  $F_1$  will contain the most important objectives and  $F_n$  the least important ones. Each family includes hypotheses corresponding to analyses at the same level of hierarchy. The



order of the families needs to be predefined in the protocol before starting the clinical trial. In vaccine context, we talk about primary, secondary and tertiary objectives.

Then, in the strategy proposed, each family (except for the last one) serves as a gatekeeper: one must pass it to perform analyses in the next family.

An example of a gatekeeping strategy with 3 families is represented as below.



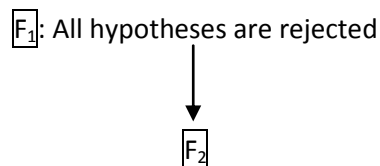
**Figure III.1:** A gatekeeping strategy with 3 families.

Here, if  $F_1$  is passed,  $F_2$  can be tested. Otherwise, neither  $F_2$  nor  $F_3$  will be tested.  $F_3$  can be tested only if  $F_1$  and  $F_2$  are passed. The arrows between the families represent this sequence and the order of how the tests are performed.

There are 3 different gatekeeping strategies: the serial, the parallel and the tree-structured strategy. Each strategy has its own specificity and is used in a proper situation as we will explain it in the following sections.

### III. 2. 1. Serial Strategy

The simplest scenario for a serial strategy with two families is represented as below.



**Figure III.2:** A problem with a serial gatekeeper ( $F_1$  is a serial gatekeeper for  $F_2$ ).

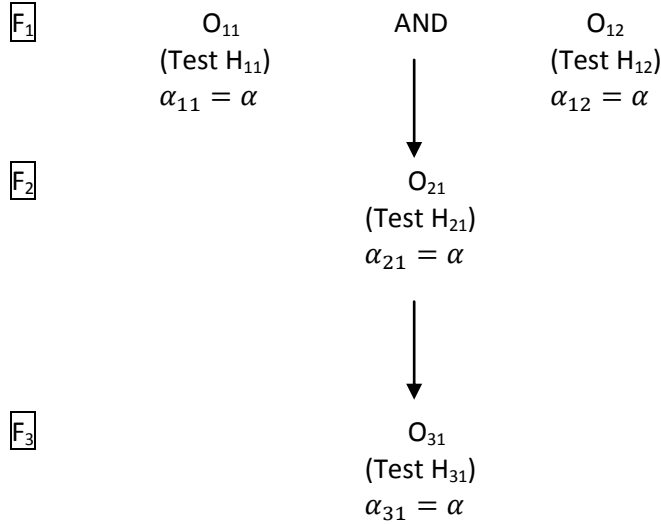
The serial strategy is used when all the objectives must be reached.

Let  $H_{11}, \dots, H_{1m_1}$  be the  $F_1$  hypotheses, with  $m_1$  the number of hypotheses in  $F_1$ .

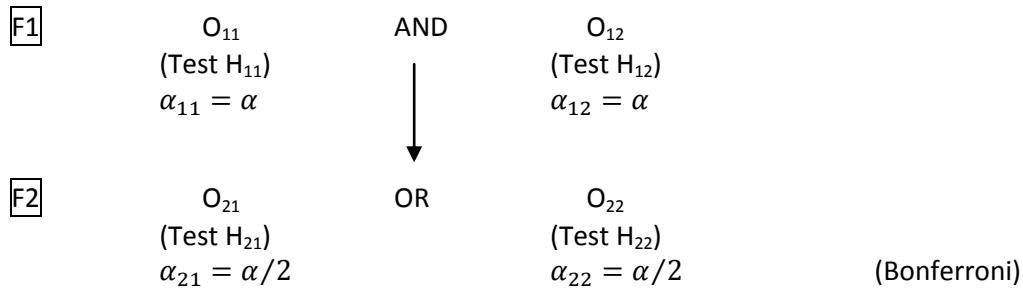
Testing  $F_1 \Leftrightarrow$  Testing  $H_{11}$  AND ... AND  $H_{1m_1}$ .

It is widely used in clinical trials, mainly due to the fact that it does not require any adjustment for multiplicity.

Let us see an example of two serial scenarios with four ordered objectives.



**Figure III.3:** Scenario 1, serial gatekeeping strategy.

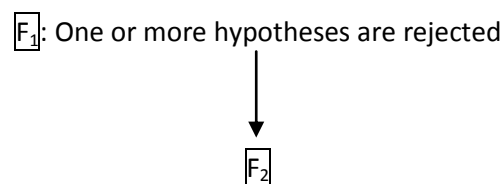


**Figure III.4:** Scenario 2, serial gatekeeping strategy with a Bonferroni adjustment in  $F_2$ .

In the family  $F_2$  of the scenario 2, we used the Bonferroni adjustment. In the serial strategy, we are allowed to use an “or” between different objectives only in the last family.

### III. 2. 2. Parallel Strategy

The principle of the parallel strategy with two families is represented as it follows.



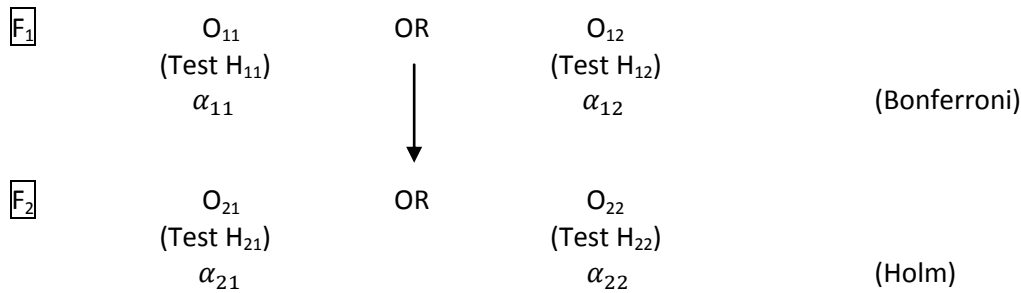
**Figure III.5:** A problem with a parallel gatekeeper ( $F_1$  is a parallel gatekeeper for  $F_2$ ).

Parallel gatekeepers are used in clinical trials with several primary objectives where each objective defines a successful trial outcome. In other words, each primary objective is independent and the trial will be positive if at least one primary objective is reached.

Let  $H_{11}, \dots, H_{1m_1}$  be the  $F_1$  hypotheses, with  $m_1$  the number of hypotheses in  $F_1$ .

Testing  $F_1 \Leftrightarrow$  Testing  $H_{11}$  OR ... OR  $H_{1m_1}$ .

This strategy requires adjustment to preserve the FWER. Let see that in an example.



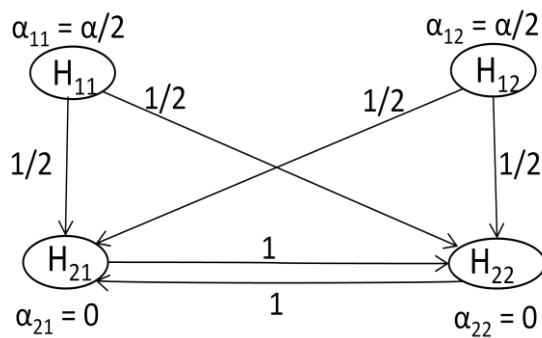
**Figure III.6:** Scenario 3, parallel gatekeeping strategy with a Bonferroni adjustment in  $F_1$  and a Holm adjustment in  $F_2$ .

Here, the adjustment in  $F_1$  is done with the Bonferroni method and with the Holm method in  $F_2$ .

To explain how the alpha level for each hypothesis is calculated, we will use the graphical illustration of the gatekeeping procedure introduced by F. Bretz et al. (cf. Reference [1]).

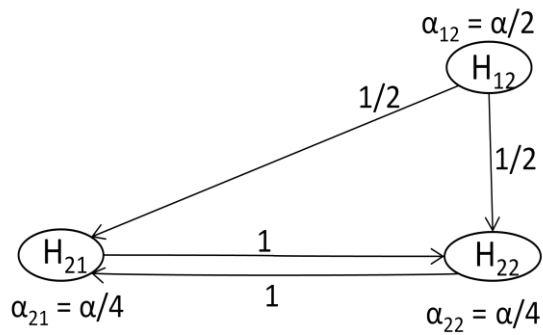
In the following graphical illustration, we will assume that the hypotheses are rejected in this order:  $H_{11}$ ,  $H_{21}$ ,  $H_{12}$  and  $H_{22}$ , because of the p-values order. The weight associated to each arrow depends on the adjustment method used (here, Bonferroni for  $F_1$  and Holm for  $F_2$ ).

(1) Hypotheses and adjusted  $\alpha$  level



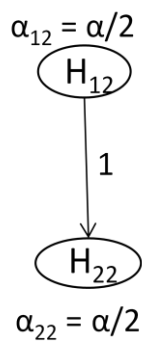
$H_{11}$  and  $H_{12}$  can be tested at level  $\alpha/2$ .  $H_{21}$  and  $H_{22}$  cannot be tested before rejecting  $H_{11}$  or  $H_{12}$ , that is why  $\alpha_{21}$  and  $\alpha_{22}$  are equal to 0.

(2) After rejection of  $H_{11}$



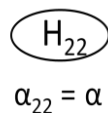
*$H_{11}$  has been rejected, so his alpha level is split and distributed to  $H_{21}$  and  $H_{22}$ .*

(3) After rejection of  $H_{21}$



*$H_{21}$  has been rejected, his alpha level is passed entirely to  $H_{22}$  as the weight between  $H_{21}$  and  $H_{22}$  was 1.*

(4) After rejection of  $H_{12}$



*$H_{12}$  has been rejected, his alpha level is passed entirely to  $H_{22}$  as the weight between  $H_{12}$  and  $H_{22}$  was 1. Now,  $H_{22}$  can be tested at level  $\alpha$ .*

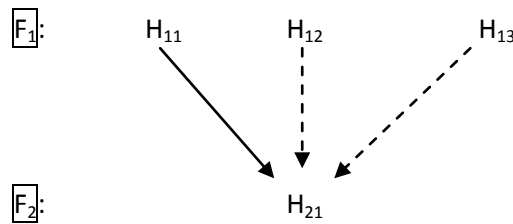
**Figure III.7:** Graphical illustration of the scenario 3, parallel strategy.

This graphical representation can be used for all strategies, but we give only an example for parallel strategy here. This representation is useful in clinical trials context as it is more understandable than numbers tables.

### III. 2. 3. Tree-structured Strategy

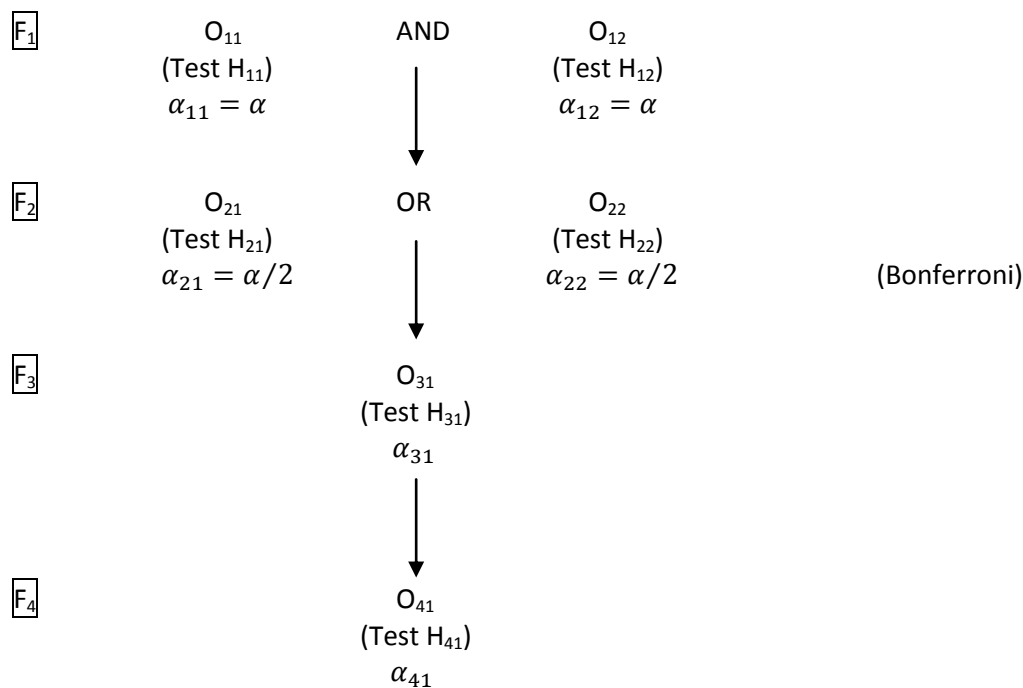
The tree-structured strategy is a combination of serial and parallel strategies. This framework is quite general and it can be applicable to a wide variety of clinical trials cases for addressing multiplicity issues.

As no scenario of this kind is used for the simulations in the next section, we will only explain the general concept of this strategy.



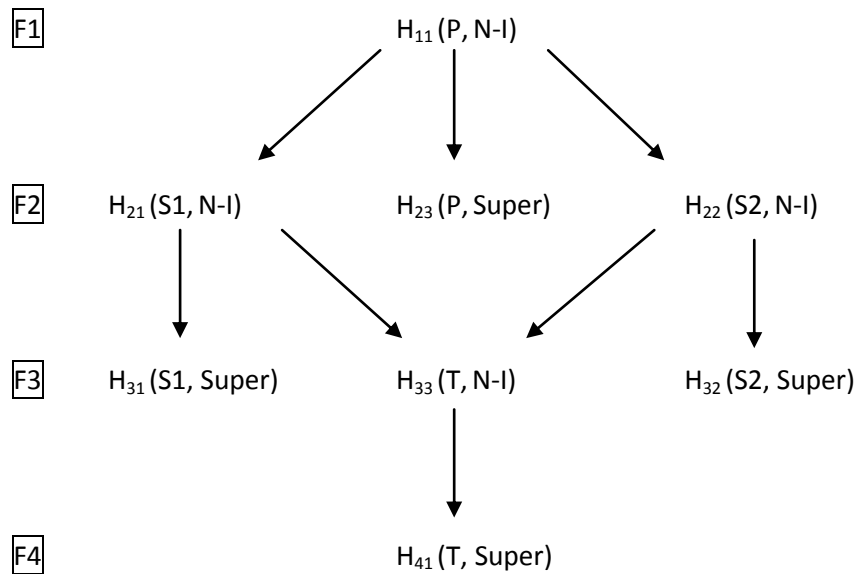
**Figure III.8:** Tree gatekeeping strategy in a two-family problem. A solid line is used to define a “serial” connection and dotted lines are used for “parallel” connections.

A simple example of a tree-structured strategy with four families is given in the next figure.



**Figure III.9:** Scenario 4, simple tree-structured gatekeeping strategy with a Bonferroni adjustment in  $F_2$ . The adjustments for alpha in  $F_3$  and  $F_4$  are the same as before in the parallel strategy (see Figure III.6).

Usually, tree-structured gatekeepers are defined at the hypothesis rather than family level. In other words, a hypothesis in a certain family may be testable whereas another hypothesis in the same family may not. The next figure gives us an explicit example of this fact.



**Figure III.10:** Complex tree-structured scenario example. The diagram includes references to the primary (P), two secondary (S1 and S2) and tertiary (T) endpoints as well as non-inferiority (N-I) and superiority (Super) analyses. See details given by Alex Dmitrienko et al. (cf. Reference [5], in Figure 1, p. 2471).

## IV. Simulations

As an application of the gatekeeping strategies, we will evaluate performance of different scenarios, serial or parallel, with multiple objectives.

First, we will simulate data starting from real study parameters, this data representing a dataset. Then, we will incorporate this dataset in one scenario and see which objectives are reached and which families are passed. We will obtain a score for each family.

We will repeat this process a certain number of times with the same scenario.

To do that, several datasets will be simulated, as many as the simulations we want to perform. By summing scores of all simulations and for each family and expressing it in percentage, we will evaluate a performance for the scenario in question.

Moreover, to observe performance variations in case of correlated data, the same scenario will be tested with different groups of datasets, each group associated to one correlation case.

Comparisons of performance will be made between these correlation cases for each scenario. The different scenarios will not be compared between them as they do not release same objectives and same conclusions.

All these steps will be developed in this section.

### IV. 1. Settings

This work wanted to be as realistic as possible in order to be reused on cases usually met in clinical trials. That is why we used data parameters from a real study.

#### IV. 1. 1. Study Explanation

We are interested in a study comparing two vaccines against the HPV. To compare these vaccines, the immunological response is evaluated.

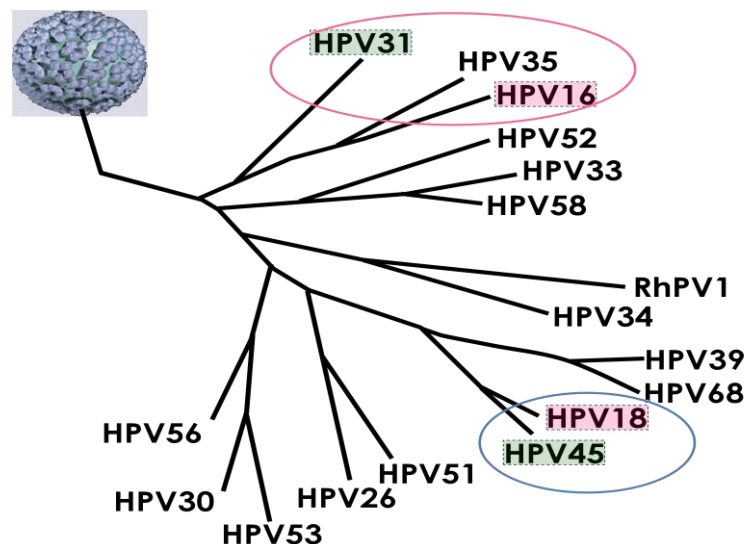
We note vaccine A for the new vaccine and vaccine B for the comparator. Subjects involved in the trial are divided in two groups. We assume here that all subjects are independent and equally distributed in the two groups. Subjects in the first group are vaccinated with the vaccine A and the ones in the second group with vaccine B.

To compare A and B, we focus on the immunological response regarding the antigen<sup>2</sup> types 16, 18, 31, 45, 6 and 11, seven months after the vaccination.

We are interested in types 16 and 18 because together these virus types account for 70 per cent of cervical cancer cases globally. The types 6 and 11 are responsible for genital warts but not for cancer.

The antigens for the types 16 and 18 are present in treatment A and B. The types 6 and 11 are only in treatment B.

The types 31 and 45 are present in none of the two treatments. Indeed, precedent studies showed that people protected against the type 16 were also protected against 31, the same for 18 and 45. So, there is probably a correlation between types 16 and 31 in one hand and between 18 and 45 in the other hand. The picture of the HPV representation above can give an explanation of that fact.



**Picture IV.1:** HPV representation with different antigen types. We note that antigen types 16 and 31, 18 and 45 are closed.

For this study, two methods were used to evaluate the immunological response.

A titre is a way to measure the number of antibodies<sup>3</sup> in the blood. The Geometric Mean of the Titres (GMT) is calculated in each group of treatment on all subjects (see section IV.3.1. for mathematical writing of Geometric Mean).

<sup>2</sup> Antigens are present in vaccines to provoke the immunological response.

<sup>3</sup> An antibody is a protein that identifies and neutralizes antigens specifically.



The number of T-cell<sup>4</sup> in the blood is another way to measure the immunological response. The Geometric Mean of the T-cell is also calculated in each group of treatment on all subjects.

In the following, we will note GM if we talk about the Geometric Mean of measures regardless of the method used.

#### **IV. 1. 2. Objectives**

This study is a complex study because several objectives need to be reached. These objectives are hierarchically ordered as it follows.

##### Primary objectives:

- Superiority of A over B for the antigen type 16 in term of post vaccination GMT.
- Superiority of A over B for the antigen type 18 in term of post vaccination GMT.

##### Secondary objectives:

- Non-inferiority of A over B for the antigen type 31 in term of post vaccination GMT.
- Non-inferiority of A over B for the antigen type 45 in term of post vaccination GMT.

##### Tertiary objectives:

- Non-inferiority of A over B for the antigen type 6 in term of GM of T-cell.
- Non-inferiority of A over B for the antigen type 11 in term of GM of T-cell.

#### **IV. 1. 3. Hypotheses Tested**

The two sorts of test are the superiority and the non-inferiority (*as seen in section II. 2. 1.*) of treatment A over treatment B. For each of them, we give the formal mathematical writing using ratios and also the criteria taken into account for the reject of null hypotheses.

We note as  $m_A$  the GM value for treatment A and  $m_B$  for treatment B.

##### Superiority

The superiority test, seen as a one-sided test is given below.  $H_0$  is rejected if a significant difference between the two treatments in favour of A is showed.

---

<sup>4</sup> A T-cell belongs to a group of white blood cells and plays a central role in immunological response.

$$H_0: \frac{m_B}{m_A} \geq 1 \quad vs. \quad H_1: \frac{m_B}{m_A} < 1$$

$$\Leftrightarrow^5 \quad H_0: \log m_B - \log m_A \geq 0 \quad vs. \quad H_1: \log m_B - \log m_A < 0$$

The statistical test used here is the unpaired t-test, as the two samples (formed with vaccines A and B) are independent and identically distributed. By performing this test, we obtain a p-value which is significant if a difference between  $\log m_B$  and  $\log m_A$  is observed. Then, it is still to verify that this difference is strictly smaller than 0 and so, we can reject  $H_0$  (cf. Appendixes B and C, p. 42 and 45, for SAS code).

#### Non-inferiority

The non-inferiority test, seen also as a one-sided test is given below.  $H_0$  is rejected if  $m_B$  is significantly smaller than  $2m_A$ . This is called the two-folded criteria, the equivalence born is noted as  $\Delta$  and equals to 2.

$$H_0: \frac{m_B}{m_A} \geq \Delta \quad vs. \quad H_1: \frac{m_B}{m_A} < \Delta \quad (\text{with } \Delta = 2)$$

$$\Leftrightarrow^6 \quad H_0: \log m_B - \log m_A \geq 0.301 \quad vs. \quad H_1: \log m_B - \log m_A < 0.301$$

The statistical test used here is also the unpaired t-test. By performing this test, we obtain a p-value which is significant if a difference between  $\log m_B$  and  $\log m_A$  not equal to 0.301 (or  $\log 2$ ) is observed. Then, it is still to verify that this difference is strictly smaller than 0.301 and so, we can reject  $H_0$  (cf. Appendixes C and F, p. 45 and 55 for SAS code).

## **IV. 2. Gatekeeping Scenarios**

The scenarios used for the simulations are as it follows. The scenarios 1 and 2 are serial strategies and the scenario 3 is a parallel one. The FWER is fixed at 0.05 for each scenario.

These scenarios were designed by using some ongoing vaccine projects.

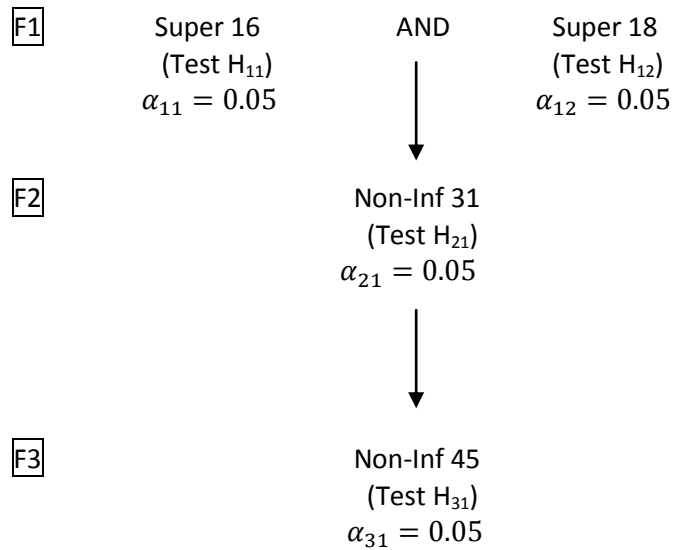
The SAS macros corresponding to scenarios used for the simulations are given in the Appendixes B, C and F, p. 42, 45 and 55.

---

<sup>5</sup> See in section IV. 3. 1. why the logarithm function is used here.

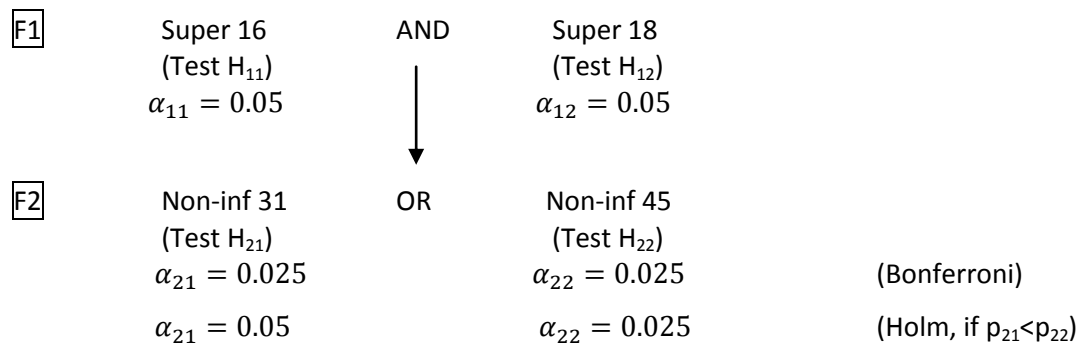
<sup>6</sup> Same remark as in <sup>5</sup>.

### Scenario 1



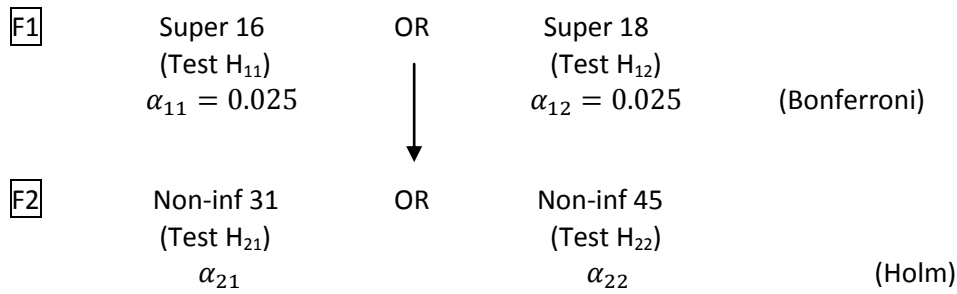
**Figure IV.2:** Scenario 1, serial gatekeeping strategy with 3 families. The diagram includes references to non-inferiority (Non-inf) and superiority (Super) analyses.

### Scenario 2



**Figure IV.3:** Scenario 2, serial gatekeeping strategy with a Bonferroni or Holm adjustment in  $F_2$ . The diagram includes references to non-inferiority (Non-inf) and superiority (Super) analyses.

### Scenario 3



**Figure IV.4:** Scenario 3, parallel gatekeeping strategy with a Bonferroni adjustment in  $F_1$  and a Holm one in  $F_2$ . The diagram includes references to non-inferiority (Non-inf) and superiority (Super) analyses.

### IV. 3. Simulated Data

As the data for each subject involved in the study are confidential, we do simulations starting from parameters of this study to obtain a dataset of similar data. The SAS code for simulations is given in the Appendixes A and D, p. 38 and 50.

#### IV. 3. 1. Normal Distribution

Normal distributed data are needed to perform statistical tests. Although, we assume that normal distribution is obtained by using logarithm function on data, as usually done in clinical trials context. Let us note  $X_j^{(i)}$  as the measure value done on the  $i^{\text{th}}$  subject (with  $i$  from 1 to  $n$ ) for the antigen type  $j$  (with  $j$  equals to 16, 18, 31, 45, 6 or 11). By using the logarithm function, we obtain  $Y_j^{(i)}$ , normally distributed, as:

$$Y_j^{(i)} = \log X_j^{(i)} \sim N(\mu_j, \sigma_j^2)$$

with  $\mu_j$  the mean and  $\sigma_j$  the standard deviation for the type  $j$ .

However, with the study data, we had values of the GM and the standard deviation  $\sigma$  for each type  $j$ . Thus, we were able to determine the parameters of the Y distribution.

$$\mu_j = \log(GM_j) = \frac{1}{n} \sum_{i=1}^n \log(X_j^{(i)}), \quad \text{as } GM_j = \sqrt[n]{\prod_{i=1}^n X_j^{(i)}}$$

And so, the parameters for each group of vaccine (A and B) and for each type in the study are as it follows:

Type 16

$$A \sim N(4.566, 0.517^2)$$

$$B \sim N(4.002, 0.476^2)$$

Type 18

$$A \sim N(4.217, 0.502^2)$$

$$B \sim N(3.354, 0.562^2)$$

Type 31

$$A \sim N(2.335, 0.760^2)$$

$$B \sim N(2, 0.525^2)$$

Type 45

$$A \sim N(1.522, 0.444^2)$$

$$B \sim N(1.42, 0.137^2)$$

Type 6

$$A \sim N(1.793, 0.6^2)$$

$$B \sim N(2.438, 0.4^2)$$

Type 11

$$A \sim N(2.304, 0.6^2)$$

$$B \sim N(2.654, 0.4^2)$$

In certain cases, we chose to modify the computed parameters values to have more interesting results with our tests (for example, with the type 31, the mean for vaccine B has been changed). Moreover, standard deviations for types 6 and 11 have been fixed by us because the real parameters have not been evaluated in the study.

With these parameters, we could simulate a dataset of measures for n subjects in each group, each subject having measure for each antigen type.

This dataset for a specific group can be represented as a matrix. Let us consider Y, the matrix of data, as:

$$Y = \begin{pmatrix} Y_{16}^{(1)} & Y_{16}^{(2)} & \dots & \dots & \dots & Y_{16}^{(n)} \\ Y_{18}^{(1)} & \ddots & \dots & \dots & \dots & \vdots \\ Y_{31}^{(1)} & \vdots & \ddots & \dots & \dots & \vdots \\ Y_{45}^{(1)} & \vdots & \vdots & \ddots & \dots & \vdots \\ Y_6^{(1)} & \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{11}^{(1)} & \dots & \dots & \dots & \dots & Y_{11}^{(n)} \end{pmatrix} = \begin{pmatrix} Y_{16} \\ Y_{18} \\ Y_{31} \\ Y_{45} \\ Y_6 \\ Y_{11} \end{pmatrix}$$

As said previously, several datasets will be simulated in order to be incorporated into scenarios and to evaluate the performance. Each dataset will be normal distributed and its parameters will be as it follows.

$$Y \sim N \left( \begin{pmatrix} \mu_{16} \\ \mu_{18} \\ \mu_{31} \\ \mu_{45} \\ \mu_6 \\ \mu_{11} \end{pmatrix}, \begin{pmatrix} \sigma_{16}^2 & \sigma_{16,18} & \sigma_{16,31} & \sigma_{16,45} & \sigma_{16,6} & \sigma_{16,11} \\ \sigma_{16,18} & \sigma_{18}^2 & \sigma_{18,31} & \sigma_{18,45} & \sigma_{18,6} & \sigma_{18,11} \\ \sigma_{16,31} & \sigma_{18,31} & \sigma_{31}^2 & \sigma_{31,45} & \sigma_{31,6} & \sigma_{31,11} \\ \sigma_{16,45} & \sigma_{18,45} & \sigma_{31,45} & \sigma_{45}^2 & \sigma_{45,6} & \sigma_{45,11} \\ \sigma_{16,6} & \sigma_{18,6} & \sigma_{31,6} & \sigma_{45,6} & \sigma_6^2 & \sigma_{6,11} \\ \sigma_{16,11} & \sigma_{18,11} & \sigma_{31,11} & \sigma_{45,11} & \sigma_{6,11} & \sigma_{11}^2 \end{pmatrix} \right)$$

As represented by the variance-covariance matrix, datasets will be simulated in case of correlation. Details are given in the next section.

#### IV. 3. 2. Correlation Cases

In order to evaluate the performance in different scenarios and in different cases of correlation, we simulated also correlated data.

Initially, we assumed that the data for different types were not correlated. Then, we were interested to study results in case of correlation as certain types of antigen are correlated in the reality (saw in section IV. 1. 1.). We had the real values of correlation but by using them, we could not really draw conclusions about the effect of correlation. Finally, we chose to fix standard correlation values, the same for each couple of type. However, these correlations constitute hard conditions regarding the performance because all types are correlated together. In a result, the performance will be bigger in case of real correlations.

So, we decided to study three different correlation cases: no correlation, medium correlation and high correlation cases.

The assume correlation matrices are:

##### No correlation

$$\begin{array}{cc} & \begin{matrix} 16 & 18 & 31 & 45 \end{matrix} & & \begin{matrix} 31 & 45 & 6 & 11 \end{matrix} \\ \begin{matrix} 16 \\ 18 \\ 31 \\ 45 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

##### Medium correlation

$$\begin{array}{cc} & \begin{matrix} 16 & 18 & 31 & 45 \end{matrix} & & \begin{matrix} 31 & 45 & 6 & 11 \end{matrix} \\ \begin{matrix} 16 \\ 18 \\ 31 \\ 45 \end{matrix} & \begin{pmatrix} 1 & 0.3 & 0.3 & 0.3 \\ 0.3 & 1 & 0.3 & 0.3 \\ 0.3 & 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 0.3 & 1 \end{pmatrix} & & \begin{pmatrix} 1 & 0.3 & 0.3 & 0.3 \\ 0.3 & 1 & 0.3 & 0.3 \\ 0.3 & 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 0.3 & 1 \end{pmatrix} \end{array}$$

#### High correlation

$$\begin{matrix} & 16 & 18 & 31 & 45 \\ \begin{matrix} 16 \\ 18 \\ 31 \\ 45 \end{matrix} & \begin{pmatrix} 1 & 0.8 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 & 0.8 \\ 0.8 & 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 0.8 & 1 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} & 31 & 45 & 6 & 11 \\ \begin{matrix} 31 \\ 45 \\ 6 \\ 11 \end{matrix} & \begin{pmatrix} 1 & 0.8 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 & 0.8 \\ 0.8 & 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 0.8 & 1 \end{pmatrix} \end{matrix}$$

#### **IV. 4. Performance Computing**

For one scenario, the performance is evaluated for each family of objectives and will be the proportion of times that a gatekeeper family is passed in the scenario.

Generally in a clinical trial designs, the significance level and the power are fixed as required by the regulatory authorities and so, the sample size is calculated according to these values. In our case, we choose to fix the significance level and the sample size to analyse performance variations for the different scenarios. The way to obtain the performance value for each scenario is given below.

Algorithm used:

For one dataset:

1. Hypotheses are tested for each antigen type.
  2. If a significant p-value is obtained, count 1 and 0 otherwise.
  3. If the objectives required in the first family are reached, the gate for  $F_1$  is equal to 1.
  4. If the gate for  $F_1$  is equal to 1, the second family objectives can be tested, and the gate for  $F_2$  is equal to 1 if they are reached.
  5. Proceed in the same way for all the following families until the last one.
- ➔ Do these steps for each simulated dataset separately.
- ➔ Sum the gate values for each family.

The application of this process is given by the next figure in an example using scenario 1, with 10000 simulations.

Sim.	Type	P-val. super	Diff. super	P-val. Non-inf	Diff. Non-inf	adjusted $\alpha$	Super	Non-inf	Gate	Family
1	16	0.03	-0.49			0.05	1		1	F1
	18	<0.001	-0.89			0.05	1			
	31			<0.001	-0.54	0.05		1	1	F2
	45			0.08	0.07	0.05		0	0	F3
10000	...	...	...	...	...	...	...	...	...	...

**Figure IV.5:** Example of performance computing for scenario 1 with 1000 simulations. The p-value compared to the adjusted alpha level and the mean are needed to test superiority and non-inferiority respectively. In the table, a “1” indicates that an objective is reached, i.e. superiority or non-inferiority, or that a gate is passed.

In the first simulation, the gate for the first family is passed (i.e. is equal to 1) as the superiority was obtained for types 16 and 18. So, Non-inferiority for type 31 can be tested and is also obtained: the  $F_2$  gate is equal to 1. Non-inferiority for type 45 can be tested but is not obtained, so the  $F_3$  gate is equal to 0.

We do this for each simulation and sum the gate scores for each family. Then, we calculate a proportion with the sum of the gate scores over the number of simulations. This proportion, expressed in percentage is the performance value of the scenario.

The results tables obtained are given in the next section.



## V. Results and Analysis

### V. 1. Tables

in %	no corr.	med. corr.	high corr.
<b>F1</b>	93.85	93.59	93.65
<b>F2</b>	79.82	80.91	83.47
<b>F3</b>	76.84	78.74	82.84

**Table V.1:** Scenario 1 (serial strategy) performance results, obtained with N=20, number of subjects in each group and 10000 simulations. The performance is stated in percentage for each family of the scenario and for the three different types of correlation: no correlation (no corr.), medium correlation (med. corr.) and high correlation (high corr.)

in %	no corr.		med. corr.		high corr.	
	Holm	Bonf.	Holm	Bonf.	Holm	Bonf.
<b>F1</b>	93.85		93.59		93.65	
<b>F2</b>	92.55		91.63		90.53	
<b>both</b>	76.6	67.5	78.17	69.69	82.04	75.18

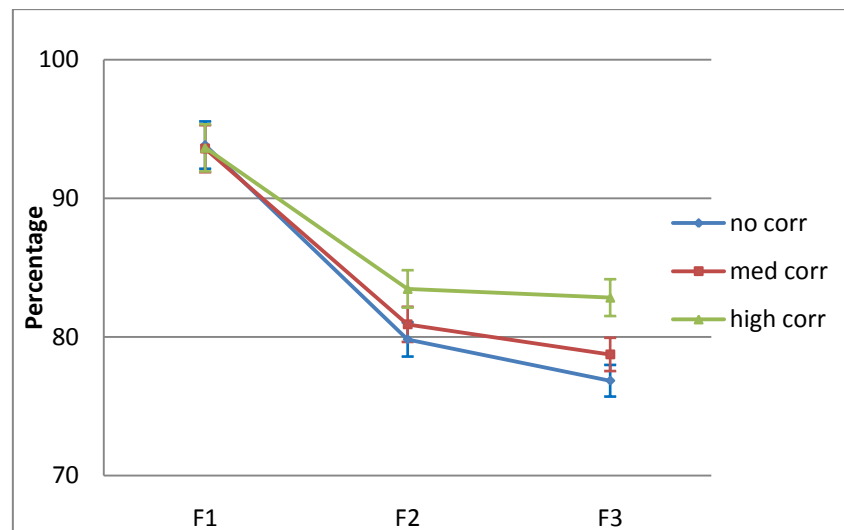
**Table V.2:** Scenario 2 (serial strategy) performance results, obtained with N=20 and 10000 simulations. The performance is stated in percentage for each family of the scenario and for the three different types of correlation: no correlation (no corr.), medium correlation (med. corr.) and high correlation (high corr.) In  $F_2$ , the performance is calculated for the first objective to be reached and then when both objectives (noted as both) are reached, with the Holm adjustment (noted as Holm) and with the Bonferroni one (noted as Bonf.)

in %	no corr.	med. corr.	high corr.
<b>F1</b>	98.5	97	93.8
<b>F2</b>	0.5	1	0.9

**Table V.3:** Scenario 3 (parallel strategy) performance results, obtained with N=20 and 1000 simulations. The performance is stated in percentage for each family of the scenario and for the three different types of correlation: no correlation (no corr.), medium correlation (med. corr.) and high correlation (high corr.)

## V. 2. Analysis

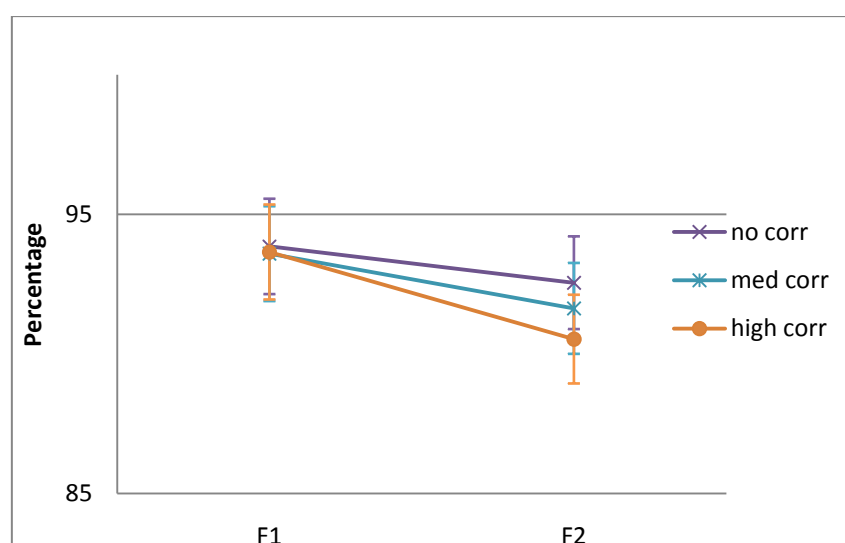
### Scenario 1



**Figure V.1:** Scenario 1, graph of performance in each family in the case of no correlation, medium and high. The 95% confidence intervals of the performance proportion are also represented.

- We observe that the correlation has no effect on the  $F_1$  performance.
- In case of high correlation, the performance is bigger in  $F_2$  and  $F_3$  than in case of no or medium correlation.
- For  $F_2$  and  $F_3$ , the performance in case of no and medium correlation are rather similar.

### Scenario 2



**Figure V.2:** Scenario 2, graph of performance in each family in the case of no correlation, medium and high. The 95% confidence intervals of the performance proportion are also represented.

- First, we can see that the performance in  $F_2$  slightly decreases when correlation between types increases but taking into account the confidence intervals, no tendency can really be showed.
- Then, we note that the Holm method obtain a better performance than the Bonferroni one, as seen in the table for  $F_2$  when both objectives are reached.

### Scenario 3

A representation graph is not useful to interpret results in this case because the number of simulations performed was not big enough and so the confidence intervals of performance are very large.

- First, we observe that the performance in  $F_1$  decreases when the correlation increases. But because of the small number of simulations performed, the confidence intervals are very large and no tendency can really be given.
- Then, we notice that the performance score for  $F_2$  is null. The objectives of non-inferiority for type 31 or type 45 with a Bonferroni adjustment have not been reached.

## VI. Conclusions and Discussion

We have seen that multiplicity causes important problems in clinical trials and that adjustments are needed.

Several multiple testing methods are available to adjust the Familywise Error Rate, as the Bonferroni and the Holm methods.

The gatekeeping method with the serial, parallel and tree-structured strategy is efficient to perform adjustments as well. This method manages multiplicity issues arising in a wide variety of applications. In clinical trials, testing multiple objectives is frequent and the gatekeeping method allows to hierarchically organize them. The use of three different strategies gives flexibility about the priority and necessity of each objective.

Using simulations, we compared performance between three cases of correlated data for three different scenarios, two serial ones and a parallel one. We found that increasing correlation has maybe a synergic effect on performance for scenario 1. No conclusions can really be drawn for the other scenarios. The results obtained are not precise because the number of simulations performed was not big enough.

This work was limited because the FWER, the sample size, the correlation and the scenarios were fixed for the simulations. For further researches, we could evaluate performance of scenarios by modifying the sample size and the FWER for example, as these three parameters are linked. A limitation was also given by the number of simulations performed. Performing more simulations gives more precision to results.

To finish, this work is only an example of the gatekeeping strategies use, given its limitations. However, it will be reused by the GSK “multiplicity project group” for their researches.

This method is still a new concept and, during my internship, I had the opportunity to present my research results in a conference to GSK statisticians based in Belgium, in the USA and in India.

The gatekeeping strategies are not yet expanded in clinical trials due to the lack of information but it could become, in few years, the most popular method of multiplicity adjustments.

## VII. References

- [1] Bretz F., Maurer W., Brannath W., Posch M. *A graphical approach to sequentially rejective multiple test procedures*. Statistics in Medicine 2009; 28:586-604.
- [2] Dmitrienko A., Offen W., Westfall P. H. *Gatekeeping strategies for clinical trials that do not require all primary effects to be significant*. Statistics in Medicine 2003; 22:2387-2400.
- [3] Dmitrienko A., Tamhane A. C. *Gatekeeping procedures with clinical trial applications*. Pharmaceutical Statistics 2007; 6: 171-180
- [4] Dmitrienko A, Tamhane A. C., Bretz F. *Multiple testing problems in pharmaceutical statistics*. Chapman & Hall 2009. CRC Press.
- [5] Dmitrienko A., Wiens B., Tamhane A. C., Wang X. *Tree-structured Gatekeeping tests in clinical trials with hierarchically ordered multiple objectives*. Statistics in Medicine 2007; 26:2465-2478.
- [6] Dmitrienko A., Molenberghs G., Chuang-Stein C., Offen W. *Analysis of Clinical Trials Using SAS: A Practical Guide*. SAS (Chapter 2).
- [7] Turk D. C., Dworkin R. H., McDermott M. P., Bellamy N. et al. *Analyzing multiple endpoints in clinical trials of pain treatments: IMMPACT recommendations*. Pain 139 (2008); 485-493.
- [8] Westfall P. H., Wolfinger R. D. *Closed Testing Procedures and PROC MULTTEST*. SAS Institute Inc. 2000, Cary, North Carolina, USA.

## VIII. Appendixes

<b><u>Appendix A:</u></b> SAS Macro for simulations of datasets for scenarios 1 and 2.....	<b>38</b>
<b><u>Appendix B:</u></b> SAS Macro for testing scenario 1 (serial).....	<b>42</b>
<b><u>Appendix C:</u></b> SAS Macro for testing scenario 2 (serial).....	<b>45</b>
<b><u>Appendix D:</u></b> SAS Macro for simulations of datasets for scenario 3.....	<b>50</b>
<b><u>Appendix E:</u></b> SAS Macro for testing objectives of scenario 3.....	<b>54</b>
<b><u>Appendix F:</u></b> SAS Macro for computing performance of scenario 3 (parallel).....	<b>55</b>

## Appendix A: SAS Macro for simulations of datasets for scenarios 1 and 2

```
%macro simulation_correl(sim, n, cor);

* Group A;
data c (drop = group);
    group="A";
    col1=0;
    col2=0;
    col3=0;
    col4=0;
    if group="B";
run;

%macro sim;
    %do i=1 %to &sim.;
        Proc IML;
            reset NoLog NoPrint;
            * Specify random number seed;
            seed=0;
            * Specify number of observations for the data matrix;
            n=&n.;
            * Variance matrix;
            Ds=Diag({0.517 0.502 0.760 0.444});
            *Covariance Matrix;
            **no correlation case;
            if &cor.=0 then
                S=Ds*Ds;
            **medium correlation case (with 0.3 correlation between
            each type);
            if &cor.=2 then do;
                R={1 0.3 0.3 0.3, 0.3 1 0.3 0.3, 0.3 0.3 1 0.3,0.3
                    0.3 0.3 1};
                S=Ds*R*Ds;
            end;
            **high correlation case (with 0.8 correlation between
            each type);
            if &cor.=3 then do;
                R={1 0.8 0.8 0.8, 0.8 1 0.8 0.8, 0.8 0.8 1 0.8, 0.8
                    0.8 0.8 1};
                S=Ds*R*Ds;
            end;
            * Compute Choleski Root for transformation;
            T=Root(S);
            * Initialize data vector with random number seed;
            X=J(n,NRow(S),Seed);
            * Generate Independent Standard Normals for the data
            matrix;
            X=Rannor(X);
            * compute the mean vector into a matrix(10,4);
            mu = J(&n.,4,0);
            do i=1 to &n.;
                mu[i,1] = 4.566;
                mu[i,2] =4.217;
                mu[i,3] =2.335;
                mu[i,4] =1.522;
            end;
            * Now transform to have the desired covariance structure;
            Y=X*T+mu;
            * Save the data matrix;
            Create c&i From Y;
        end;
    end;
%end;
```

```

        Append From Y;
        Close c&i;
Quit;

data c&i;
    set c&i;
    sim=&i;
    pid=_N_+&n.;
    group="A";
run;

data c;
    set c c&i.;
run;
PROC SQL;
    drop table c&i;
QUIT;
%end;
%MEND;
%sim;

data C16;
    set c;
    type=16;
    res=col1;
    keep sim pid group type res;
run;
data C18;
    set c;
    type=18;
    res=col2;
    keep sim pid group type res;
run;
data C31;
    set c;
    type=31;
    res=col3;
    keep sim pid group type res;
run;
data C45;
    set c;
    type=45;
    res=col4;
    keep sim pid group type res;
run;

data A;
    set C16 C18 C31 C45;
run;
**end group A;

** Group B;
data g (drop = group);
    group="B";
    col1=0;
    col2=0;
    col3=0;
    if group="A";
run;

```



```

%macro sim2;
  %do i=1 %to &sim.;
    proc iml;
      Reset NoLog NoPrint;
      * Specify random number seed;
      seed=0;
      * Specify number of observations for the data matrix;
      n=&n.;
      * Variance matrix;
      Ds=Diag({0.476 0.562 0.525 0.137});
      *Covariance Matrix;
      **no correlation case;
      if &cor.=0 then
        S=Ds*Ds;
      **medium correlation case (with 0.3 correlation between
      each type);
      if &cor.=2 then do;
        R={1 0.3 0.3 0.3, 0.3 1 0.3 0.3, 0.3 0.3 1 0.3,0.3
          0.3 0.3 1};
        S=Ds*R*Ds;
        end;
      **high correlation case (with 0.8 correlation between
      each type);
      if &cor.=3 then do;
        R={1 0.8 0.8 0.8, 0.8 1 0.8 0.8, 0.8 0.8 1 0.8, 0.8
          0.8 0.8 1};
        S=Ds*R*Ds;
        end;
      if &cor.=4 then do;
        R={1 0.8 0 0, 0.8 1 0 0, 0 0 1 0, 0 0 0 1};
        S=Ds*R*Ds;
        end;
      * Compute Choleski Root for transformation;
      T=Root(S);
      * Initialize data vector with random number seed;
      X=J(n,NRow(S),Seed);
      * Generate Independent Standard Normals for the data
      matrix;
      X=Rannor(X);
      * compute the mean vector into a matrix(10,4);
      mu = J(&n.,4,0);
      do i=1 to &n.;
        mu[i,1] = 4.002;
        mu[i,2] =3.354;
        mu[i,3] =2;
        mu[i,4] =1.42;
      end;
      * Now transform to have the desired covariance structure;
      Y=X*T+mu;

      * Save the data matrix;
      Create g&i From Y;
      Append From Y;
      Close g&i;
    Quit;

    data g&i;
      set g&i;
      sim=&i;
      pid=_N_;
      group="B";
  run;

```

```

        data g;
            set g g&i.;
        run;
        PROC SQL;
            drop table g&i;
        QUIT;
    %end;
%MEND;
%sim2;

data G16;
    set g;
    type=16;
    res=col1;
    keep sim pid group type res;
run;

data G18;
    set g;
    type=18;
    res=col2;
    keep sim pid group type res;
run;

data G31;
    set g;
    type=31;
    res=col3;
    keep sim pid group type res;
run;

data G45;
    set g;
    type=45;
    res=col4;
    keep sim pid group type res;
run;

data B;
    set G16 G18 G31 G45;
run;
**end of group B;

data res.N&n._&cor.corr;
    set A B;
run;

proc sort data=res.N&n._&cor.corr;
    by sim type pid;
run;

%mend;

%simulation_correl(sim=10000, n=20, cor=0);
%simulation_correl(sim=10000, n=20, cor=2);
%simulation_correl(sim=10000, n=20, cor=3);

```

## **Appendix B: SAS Macro for testing scenario 1 (serial)**

```
/*Scenario1 :

    16(sup) AND 18(sup) --> 31(non-inf) --> 45(non-inf)

    A vs. B */

%macro scenariol(sim, n, c);

/**two independent samples t-test***/

ods listing close;
proc ttest data=res.N&n._&c.corr alpha=0.05 order=data;
    class group;
    by sim type;
    var res;
    ods output Ttests=TG1; *to obtain p value;
    ods output statistics=TG1_stat; *to obtain mean and upper limit;
run;
ods listing;

data TG_stat;
    set TG1_stat;
    keep sim type mean upperclmean;
    where class contains 'Diff';
run;

/**read-out format;*/

proc format;
    value adj
        low-<0.0001='<.0001';
run;

/**TG3 = ttest B reduced (we assume the 2 samples have same variance)***/

data TG2;
    set TG1;
    where method contains 'Pooled';
    keep sim type probt;
run;

data TG;
    merge TG2 TG_stat;
    by sim type ;
    if type=16 or 18 then adj_p=probt; /*formula for adjusted p here*/
    if type=31 or 45 then adj_p=probt;
    format adj_p adj.;
    if (adj_p<0.05 and mean<0) then superiority=1; *comparer p a 0.05 ou
0.025 ? (2sided);
        else superiority=0;
    if Upperclmean<0.301 then noninferiority=1;
        else noninferiority=0;
run;

proc sort data=TG;
    by type;
```

```

run;

data res.S1_TG&n.N_&c.corr;
    set tg;
    run;

proc means data=TG sum;
    var superiority noninferiority;
    by type;
    output out=summary1 sum(superiority noninferiority)=sup noninf ;
run;

data summary2;
    N=&n.;
    do type=16,18,31,45;
        output;
    end;
run;

/**performance for each objective**/

data res.S1P_&n.N_&c.corr;
    merge summary2 summary1; *set summary2;
    by type;
    perf_sup1=(sup/&sim.)*100;
    perf_noninf1=(noninf/&sim.)*100;
    format perf_sup perf_noninf $8.;
    perf_sup=left(trim(perf_sup1) || "%");
    perf_noninf=left(trim(perf_noninf1) || "%");
    keep N type perf_sup perf_noninf;
run;

/** performance computing **/

proc sort data=TG;
    by sim;
run;

/*test S*/

data TG_P;
    set TG;
    gate=0;
    if type = 16 and superiority = 1 then gate = 1;
    if type = 18 and superiority = 1 and lag(gate) = 1 then gate =1;
    *test si sup pr 16 ou 18 : if type in (16,18) and superiority = 1
then gate = 1;
    if type = 31 and noninferiority = 1 and lag(gate) = 1 then gate = 1;
    if type = 45 and noninferiority = 1 and lag(gate) = 1 then gate = 1;
run;

proc sort data=TG_P;
    by type;
run;

data overall_perfor1;
    set TG_P;
    if type=16 then delete;
    keep sim type gate;
run;

proc sort data=overall_perfor1;

```

```

        by type;
run;

proc means data=overall_perfor1 sum;
    var gate;
    by type;
    output out=overall_perfor sum(gate)=perfor1 ;
run;

data overall_perfor2;
    N=&n.;
    do family="F1", "F2", "F3";
        if family="F1" then type=18;
        if family="F2" then type=31;
        if family="F3" then type=45;
    output;
    end;
run;

data res.S1F_&n.N_&c.corr;
    merge overall_perfor2 overall_perfor;
    by type;
    perfor1=(perfor1/&sim.)*100;
    format perfor $8.;
    perfor=left(trim(perfor1) || "%") ;
    drop _FREQ_ _TYPE_ type perfor1;
run;

%mend;

%scenario1(sim=10000,n=20, c=0);
%scenario1(sim=10000,n=20, c=2);
%scenario1(sim=10000,n=20, c=3);

```

## Appendix C: SAS Macro for testing scenario 2 (serial)

```
/*Scenario2 :

    16(sup) AND 18(sup) --> 31(non-inf) OR 45(non-inf) with Bonferroni
and Holm ajustements

    A vs. B */

%macro scenario2(sim, n, c);

/**two independent samples t-test***/

ods listing close;
/*ttest for the superiority hypotheses*/
proc ttest data=res.N&N._&c.corr alpha=0.05 order=data;
    class group;
    by sim type;
    var res;
    ods output Ttests=TG1sp; *to obtain p value;
    ods output statistics=TG1sm; *to obtain mean;
run;
ods listing;

ods listing close;
/*ttest for the non-inferiority hypotheses*/
proc ttest data=res.N&N._&c.corr alpha=0.05 order=data H0=0.301;
    class group;
    by sim type;
    var res;
    ods output Ttests=TG1nip; *to obtain p value for non inf;
    ods output statistics=TG1nim; *to obtain mean;
run;
ods listing;

/**read-out format;*/

proc format;
    value adj
        low-<0.0001='<.0001';
run;

data TG2sp;
    set TG1sp;
    where method contains 'Pooled';
    keep sim type probt;
run;

/* sorting p-values for type 31 and 45 : */

proc sort data=TG1nip;
    by sim descending type;
    where method contains 'Pooled';
run;

data TG2nip;
    set TG1nip;
    temp=lag(probt);
    if type=31 then
        if temp>=probt then ordre=1;
```

```

                                else ordre=2;
        p_noninf=probt;
        keep sim type p_noninf ordre;
run;

proc sort data=TG2nip;
    by sim type;
run;

data TG22nip;
    set TG2nip;
    if type=45 and lag(ordre)=2 then ordre=1;
    if type=45 and lag(ordre)=1 then ordre=2;
run;

/*end of sorting*/

data TG2sm;
    set TG1sm;
    keep sim type mean;
    where class contains 'Diff';
run;

data TG2nim;
    set TG1nim;
    mean_noninf=mean;
    keep sim type mean_noninf;
    where class contains 'Diff';
run;

data TGm;
    merge TG2sp TG2sm TG22nip TG2nim;
    by sim type ;
    noninf_hom=0;
run;

proc sort data=TGm;
    by sim ordre type;
run;

data TG;
    set TGm;
    p_val=probt;
    if type in (16,18) then

                                if (p_val<0.05 and mean<0) then
                                    superiority=1;
                                else superiority=0;

                                do;
                                    *Bonferroni ajustement : ;
                                    adj_Bonf=2*p_noninf;
                                    if (adj_Bonf<0.05 and

mean_noninf<0.301) then

                                else noninf_Bonf=0;
                                *Holm ajustement : ;
                                adj_Hom=2*p_noninf/ordre;
                                if (ordre=1 and

                                noninf_Bonf=1;

adj_Hom<0.05 and mean_noninf<0.301) then

                                noninf_Hom=1;

```

```

do;
    if (ordre=2 and
        (lag(noninf_Hom))=1 and adj_Hom<0.05 and mean_noninf<0.301) then
        noninf_Hom=1;
end;

format p_val adj_Bonf adj_Hom adj.;
run;

proc sort data=TG;
    by type;
run;

data res.Data_&n.N_&c.corr;
    set TG;
run;

proc means data=TG sum;
    var superiority noninf_Bonf noninf_Hom;
    by type;
    output out=summary1 sum(superiority noninf_Bonf noninf_Hom)=sup1
noninf_Bonf1 noninf_Hom1;
run;

data summary2;
    N=&n.;
    do type=16,18,31,45;
        output;
    end;
run;

/**performance computing for each family**/

data res.S2P_&n.N_&c.corr;
    merge summary2 summary1;
    by type;
    format sup sup_Bonf sup_Hom $8.;
    if type in (16,18) then sup=left(trim((sup1/&sim.)*100) || "%");
    if type in (31,45) then
        do;

            noninf_Bonf=left(trim((noninf_Bonf1/&sim.)*100) || "%");

            noninf_Hom=left(trim((noninf_Hom1/&sim.)*100) || "%");
            end;
    keep N type sup noninf_Bonf noninf_Hom;
run;

/** overall performance **/

proc sort data=TG;
    by sim;
run;

/*test S*/

data TG1;
    set TG;
    if type = 16 and superiority = 1 then
        do;

```



```

100;

if type = 18 and superiority = 1 then

    if type in (31,45) then
        do;
            if noninf_Bonf = 1 then gate_Bonf = 1;
            if noninf_Hom = 1 and ordre=1 then gate_Hom =
1;
            if noninf_Hom = 1 and ordre=2 then
gate_Hom=0.1;
        end;
run;

proc sql;
    create table TG2 as
    select *, sum(gate_bonf) as bonf, sum(gate_hom) as hom
    from TG1
    group by sim;
quit;

proc sort data=TG2;
    by type;
run;

data TG3;
    set TG2;
    if type=18 then delete;
    if type=16 and bonf>=110 then
        do;
            perf_bonf1=1;
            perf_hom1=1;
        end;

    if type=31 then
        do;
            if bonf>110 then perf_bonf1=1;
            if hom>110 then perf_hom1=1;
        end;

    if type=45 then
        do;
            if bonf=112 then t2bonf=1;
            if hom=111.1 then t2hom=1;
        end;

    keep sim type perf_bonf1 perf_hom1 t2bonf t2hom ;
run;

proc means data=TG3 sum;
    var perf_bonf1 perf_hom1 t2bonf t2hom;
    by type;
    output out=TG4 sum(perf_bonf1 perf_hom1 t2bonf t2hom)= Bonf1 Hom1
t2bonf t2hom;
run;

data overall_perfor2;
    N=&n.;
    do family="F1 ", "F2 ", "both";
        if family="F1 " then type=16;

```

```

        if family="F2 " then type=31;
        if family="both" then type=45;
        output;
    end;
run;

data res.S2F_&n.N_&c.corr;
    merge overall_perfor2 TG4;
    by type;
    format perf_Hom perf_Bonf perf_2bonf perf_2hom $8.;
    perf_Bonf=left(trim((Bonf1/&sim.)*100) || "%");
    perf_Hom=left(trim((Hom1/&sim.)*100) || "%");
    perf_2bonf=left(trim((t2bonf/&sim.)*100) || "%");
    perf_2hom=left(trim((t2hom/&sim.)*100) || "%");
    drop type _FREQ_ _TYPE_ Bonf1 Hom1 t2bonf t2hom;
run;

%mend;

%scenario2(sim=10000,n=20,c=0);
%scenario2(sim=10000,n=20,c=2);
%scenario2(sim=10000,n=20,c=3);

```

## Appendix D: SAS Macro for simulations of datasets for scenario 3

```
%macro simulation_correl2(sim, n, cor);

* Group A;
data c (drop = group);
    group="A";
    col1=0;
    col2=0;
    col3=0;
    col4=0;
    if group="B";
run;

%macro sim;
    %do i=1 %to &sim.;
        Proc IML;
            reset NoLog NoPrint;
            * Specify random number seed;
            seed=0;
            * Specify number of observations for the data matrix;
            n=&n.;
            * Variance matrix;
            Ds=Diag({0.76 0.444 0.8 0.8});
            *Covariance Matrix;
            **no correlation case;
            if &cor.=0 then
                S=Ds*Ds;
            **medium correlation case (with 0.3 correlation between
            each type);
            if &cor.=2 then do;
                R={1 0.3 0.3 0.3, 0.3 1 0.3 0.3, 0.3 0.3 1 0.3,
                    0.3 0.3 0.3 1};
                S=Ds*R*Ds;
            end;
            **high correlation case (with 0.8 correlation between
            each type);
            if &cor.=3 then do;
                R={1 0.8 0.8 0.8, 0.8 1 0.8 0.8, 0.8 0.8 1 0.8,
                    0.8 0.8 0.8 1};
                S=Ds*R*Ds;
            end;

            * Compute Choleski Root for transformation;
            T=Root(S);
            * Initialize data vector with random number seed;
            X=J(n,NRow(S),Seed);
            * Generate Independent Standard Normals for the data
            matrix;
            X=Rannor(X);
            * compute the mean vector into a matrix(10,4);
            mu = J(&n.,4,0);
            do i=1 to &n.;
                mu[i,1] =2.335;*31;
                mu[i,2] =1.522;*45;
                mu[i,3] =1.793;*6;
                mu[i,4] =2.304;*11;
            end;
            * Now transform to have the desired covariance structure;
            Y=X*T+mu;
        end;
    end;
%end;
```

```

        * Save the data matrix;
        Create c&i From Y;
        Append From Y;
        Close c&i;
    Quit;

    data c&i;
        set c&i;
        sim=&i;
        pid=_N_+&n.;
        group="A";
    run;

    data c;
        set c c&i.;
    run;
    PROC SQL;
        drop table c&i;
    QUIT;
%end;
%MEND;
%sim;

data C31;
    set c;
    type=31;
    res=col1;
    keep sim pid group type res;
run;
data C45;
    set c;
    type=45;
    res=col2;
    keep sim pid group type res;
run;
data C6;
    set c;
    type=6;
    res=col3;
    keep sim pid group type res;
run;
data C11;
    set c;
    type=11;
    res=col4;
    keep sim pid group type res;
run;

data A;
    set C31 C45 C6 C11;
run;

**end of group A;

** Group B;

data g (drop = group);
    group="A";
    col1=0;
    col2=0;
    col3=0;
    col4=0;

```

```

        if group="B";
run;

%macro sim2;
  %do i=1 %to &sim.;
    proc iml;
      Reset NoLog NoPrint;
      * Specify random number seed;
      seed=0;
      * Specify number of observations for the data matrix;
      n=&n.;
      * Variance matrix;
      Ds=Diag({0.525 0.137 0.4 0.4});
      *Covariance Matrix;
      **no correlation case;
      if &cor.=0 then
        S=Ds*Ds;
      **medium correlation case (with 0.3 correlation between
      each type);
      if &cor.=2 then do;
        R={1 0.3 0.3 0.3, 0.3 1 0.3 0.3, 0.3 0.3 1 0.3,
          0.3 0.3 0.3 1};
        S=Ds*R*Ds;
      end;
      **high correlation case (with 0.8 correlation between
      each type);
      if &cor.=3 then do;
        R={1 0.8 0.8 0.8, 0.8 1 0.8 0.8, 0.8 0.8 1 0.8,
          0.8 0.8 0.8 1};
        S=Ds*R*Ds;
      end;

      * Compute Choleski Root for transformation;
      T=Root(S);
      * Initialize data vector with random number seed;
      X=J(n,NRow(S),Seed);
      * Generate Independent Standard Normals for the data
      matrix;
      X=Rannor(X);
      * compute the mean vector into a matrix(10,4);
      mu = J(&n.,4,0);
      do i=1 to &n.;
        mu[i,1] =2;*31;
        mu[i,2] =1.42;*45;
        mu[i,3] =2.438;*6;
        mu[i,4] =2.654;*11;
      end;
      * Now transform to have the desired covariance structure;
      Y=X*T+mu;

      * Save the data matrix;
      Create g&i From Y;
      Append From Y;
      Close g&i;
    Quit;

    data g&i;
      set g&i;
      sim=&i;
      pid=_N_;
      group="B";
  end;
%end;

```

```

run;

data g;
    set g g&i.;
run;
PROC SQL;
    drop table g&i;
QUIT;
%end;
%MEND;
%sim2;

data G31;
    set g;
    type=31;
    res=col1;
    keep sim pid group type res;
run;

data G45;
    set g;
    type=45;
    res=col2;
    keep sim pid group type res;
run;

data G6;
    set g;
    type=6;
    res=col3;
    keep sim pid group type res;
run;

data G11;
    set g;
    type=11;
    res=col4;
    keep sim pid group type res;
run;

data B;
    set G31 G45 G6 G11;
run;

**end of group B;

data res.N611_&n._&cor.corr&sim.;
    set A B;
run;

proc sort data=res.N611_&n._&cor.corr&sim.;
    by sim type pid;
run;

%mend;

%simulation_correl2(sim=1000, n=20, cor=0);
%simulation_correl2(sim=1000, n=20, cor=2);
%simulation_correl2(sim=1000, n=20, cor=3);

```

## **Appendix E: SAS Macro for testing objectives of scenario 3**

```
%macro test(data);

proc ttest data=&data. alpha=0.05 order=data H0=0.301;
  class group;
  by sim type;
  var res;
  ods output Ttests=TG1nip; *to obtain p value for non inf;
  ods output statistics=TG1nim; *to obtain mean;
run;

data TG2nim;
  set TG1nim;
  mean_noninf=mean;
  keep sim type mean_noninf;
  where class contains 'Diff';
run;

data TG2nip;
  set TG1nip;
  p_noninf=probt;
  keep sim type p_noninf;
  where method contains 'Pooled';
run;

data res.data&data.;
  merge TG2nim TG2nip;
run;

%mend;

%test(data=res.N611_20_0corr1000);
%test(data=res.N611_20_2corr1000);
%test(data=res.N611_20_3corr1000);
```

## Appendix F: SAS Macro for computing performance of scenario 3 (parallel)

```
/*Scenario3 :      31(non-inf) OR 45(non-inf) (with Bonferroni adjustment)
                  --> 6 (non-inf) OR 11(non-inf) (with Holm adjustment)
                  A vs. B */

%macro scenario3(data);

proc format;
  value adj
    low-<0.0001='<.0001';
run;
proc sort data=&data.;
  by sim descending type;
run;
data ordonne;
  set &data.;
  temp=lag(p_noninf);
  temp2=lag(mean_noninf);
  if type=6 then do;
    if mean_noninf<0.301 and temp>=p_noninf then
ordre=1;
    if mean_noninf<0.301 and temp2>=0.301 then ordre=1;
    if mean_noninf>=0.301 and temp2>=0.301 and
temp>=p_noninf then ordre=1;
    end;
    if type=6 and ordre^=1 then ordre=2;
run;

proc sort data=ordonne;
  by sim type;
run;

data ordonne2;
  set ordonne;
  lagordre=lag(ordre);
  if type=11 and lagordre=2 then ordre=1;
  if type=11 and lagordre=1 then ordre=2;
run;

data noninf1;
  set ordonne2;
  if type in (31,45) then p_adj=p_noninf*2;
  if type=31 and p_adj<0.05 and mean_noninf<0.301 then gate=100;
  if type=45 and p_adj<0.05 and mean_noninf<0.301 then gate=10;
run;

proc sql;
  create table noninf2 as
  select *, sum(gate) as Flgate
  from noninf1
  group by sim;
quit;

proc sort data=noninf2;
  by sim descending type ordre;
run;

data noninf3;
  set noninf2;
```



```

        if Flgate=0 then do;
                                Flone=0;
                                Fltwo=0;
                                end;
        if Flgate in(100,10) then Flone=1;
        if Flgate=110 then Fltwo=1;

        if ordre=1 and Flone=1 then do;
                                p_adj=p_noninf*4;
                                if p_adj<0.05 and mean_noninf<0.301
then
                                F2one=1;
                                end;
                                if ordre=1 and Fltwo=1 then do;
                                p_adj=p_noninf*2;
                                if p_adj<0.05 and mean_noninf<0.301
then
                                F2one=1;
                                end;
                                end;
run;

proc sort data=noninf3;
        by sim ordre;
run;

data noninf4;
        set noninf3;
        lF2one=lag(F2one);
        if ordre=2 and lF2one=1 then do;
                                if Flone=1 then do;
                                p_adj=p_noninf*2;
                                if p_adj<0.05 and
mean_noninf<0.301 then do;
                                F2two=1;
                                F2one=0;
                                end;
                                end;
                                if Fltwo=1 then do;
                                p_adj=p_noninf;
                                if p_adj<0.05 and mean_noninf<0.301
then do;
                                F2two=1;
                                F2one=0;

                                end;
                                end;
                                end;
run;

proc means data=noninf4 sum;
        var Flone Fltwo F2one F2two;
        by type;
        output out=TG4 sum(Flone Fltwo F2one F2two)= Flone1 Fltwo1 F2one1
F2two1;
run;
%mend;

%scenario3(data=res.data611_20_0corr1000);
%scenario3(data=res.data611_20_2corr1000);
%scenario3(data=res.data611_20_3corr1000);

```