

## **Planung und Implementierung der Auswertung einer replikativen Bioäquivalenzstudie mit Hilfe der „Scaled Average Bioequivalence“**

Feryal Tanriverdio	Arne Ring
Boehringer Ingelheim Pharma GmbH & Co. KG	Boehringer Ingelheim Pharma GmbH & Co. KG
Birkendorfer Strasse 65	Birkendorfer Strasse 65
88397 Biberach an der Riss	88397 Biberach an der Riss
Feryal.Tanriverdio@	Arne.Ring@
bc.boehringer-ingelheim.com	bc.boehringer-ingelheim.com

### **Zusammenfassung**

Im Laufe der klinischen Entwicklung eines Wirkstoffes werden meist verschiedene Formulierungen (Zusammensetzung der Tabletten, Dragees etc.) untersucht. Dabei stellt sich die Frage nach der Sicherheit und Wirksamkeit der neuen Formulierung. Diese wird durch sogenannte Bioverfügbarkeits- bzw. Bioäquivalenzuntersuchungen ermittelt.

Besitzt ein Wirkstoff eine hohe pharmakokinetische Variabilität, müssen in traditionellen Bioäquivalenzuntersuchungen sehr hohe Fallzahlen eingesetzt werden, um die Bioäquivalenz-Kriterien erfüllen zu können. Ein möglicher Ausweg ist dabei die „scaled average bioequivalence“, mit der die Bioäquivalenzgrenzen entsprechend der Variabilität der Referenzformulierung aufgeweitet werden. Diese Methode verlangt jedoch einen höheren Aufwand bei der Implementierung, da die verwendeten Statistiken nicht normalverteilt sind; spezielle Linearisierungen oder Bootstrap-Verfahren stellen mögliche Lösungsansätze dar.

In der vorliegenden Studie wurde die SABE-Methodik erstmals bei Boehringer Ingelheim eingesetzt. Trotzdem sollte die Auswertung innerhalb kurzer Zeit nach Erhalt der pharmakokinetischen Parameter in validierter Form vorliegen.

Um dies zu gewährleisten, wurden die Daten einer abgeschlossenen Bioäquivalenzstudie mit ähnlichem Design herangezogen, deren Datensätze an die Strukturen der aktuellen Studie angepasst wurden. Auf dieser Basis konnte die Programmierung der Auswertung bereits vor Vorliegen der eigentlichen Studiendaten durchgeführt werden.

Im Folgenden soll erläutert werden, welche Probleme – insbesondere bei der Bootstrap-Generierung – im Zusammenhang mit der Umsetzung in SAS<sup>®</sup> auftraten und wie diese gelöst werden konnten.

**Schlüsselwörter:** Bioäquivalenz, Frontloading, SAS<sup>®</sup> -Makros, Bootstrap

# 1 Einführung

## 1.1 Bioäquivalenznachweis

Die Bioäquivalenzanalyse stellt eine wichtige Anwendung der Untersuchungen oral applizierter Formulierungen dar. Wenn eine neue Formulierung eines Pharmakons entwickelt wurde, muss nachgewiesen werden, dass diese den bereits untersuchten (z.B. während der klinischen Entwicklung) bzw. zugelassenen (nach Ablauf der Patentlaufzeit) Produkten entspricht. Dazu werden pharmakokinetische Maßzahlen verglichen, die ausgehend von den Konzentrations-Zeit-Verläufen der bisherigen (Referenz-)Formulierung und der neuen (Test-)Formulierung bestimmt wurden. Die resultierenden Konzentrations-Zeit-Verläufe nach Applikation von Test- und Referenzformulierung müssen nicht identisch, sondern innerhalb definierter Grenzen ähnlich sein. Die Annahme ist dabei, dass bei ähnlicher Bioverfügbarkeit von Referenz- und Testprodukt auch die Sicherheit und Wirksamkeit ähnlich sind.

Die Bestimmungen zur Arzneimittelzulassung von Pharmaka sind in verschiedenen Ländern auf ähnlichen Konzepten basierend realisiert. Entsprechend des traditionellen Verfahrens (Average Bioequivalence – ABE) gelten zwei Formulierungen als bioäquivalent, wenn das 90% Konfidenzintervall des Verhältnisses der Bioverfügbarkeit von Test- und Referenzpräparat vollständig im Bereich von 80-125% liegt (two-one sided test procedure, [1]).

$$0.8 \leq \frac{g\mu_T}{g\mu_R} \leq 1.25 \quad (1)$$

Dazu werden sogenannte „Bioäquivalenz-Metriken“ als modellunabhängige Maßzahlen zum Vergleich verwendet: Die AUC (Fläche unter der Kurve) als Maßzahl des Ausmaßes und  $C_{\max}$  (Maximale Konzentration) für die Geschwindigkeit der Absorption. Für beide Parameter werden die geometrischen Mittelwerte (Average) der Test- ( $g\mu_T$ ) und Referenz-Formulierung ( $g\mu_R$ ) berechnet (die meisten pharmakokinetischen Größen werden als log-normalverteilt angenommen).

Wird Gleichung (1) logarithmiert, kann die Berechnung von Schätzer und Konfidenzintervall mit Hilfe eines linearen Modells erfolgen, da der Differenz der beiden Mittelwerte eine Normalverteilung zugrunde liegt, wobei sich die neuen Grenzen  $\ln(-\theta_A) = -0.223$  und  $\ln(\theta_A) = +0.223$  ergeben.

$$\ln(-\theta_A) \leq \ln(\mu_T) - \ln(\mu_R) \leq \ln(\theta_A) \quad (2)$$

## 1.2 Bioäquivalenzanalyse bei "Highly variable drugs"

Für Substanzen mit einer hohen pharmakokinetischen Variabilität ist das ABE-Verfahren schlecht geeignet, da zur Erzielung eines ausreichend kleinen Konfidenzintervalles eine hohe Fallzahl benötigt wird. Als hoch variabel werden Substanzen angesehen, die

eine intraindividuelle Variabilität von grösser 25- 30 % besitzen (wobei diese Grenze noch nicht allgemein verbindlich festgelegt ist).

In den letzten Jahren wurden verschiedene methodische Ansätze diskutiert, die unter Bewahrung des  $\alpha$ -Levels die Power der Studie erhöhen können. Diese Diskussion ist bereits fortgeschritten, doch gibt es noch keine verbindlichen regulatorischen Aussagen zur Akzeptanz der Methoden.

In der vorliegenden Studie wurde die „scaled average bioequivalence (SABE)“-Methode eingesetzt, deren Konzept sich an die ABE-Auswertung anlehnt, wobei die Akzeptanzgrenzen jedoch entsprechend der Varianz der Referenzformulierung aufgeweitet werden [3]:

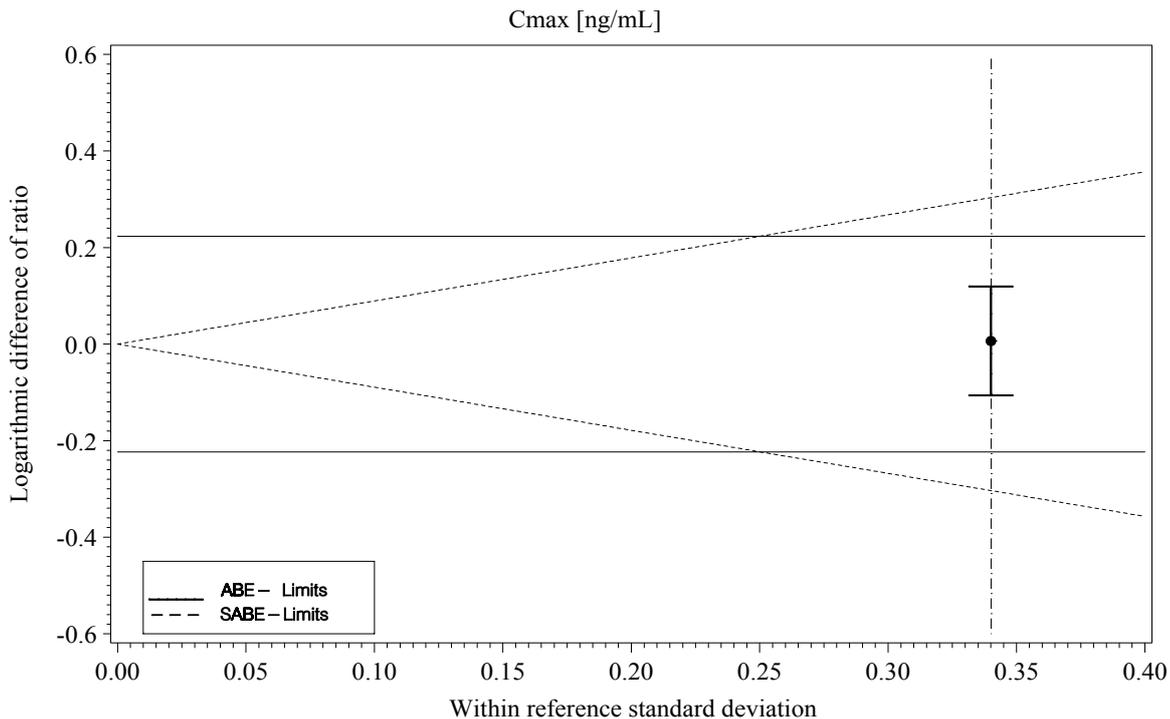
$$M_{as} = \frac{(\mu_T - \mu_R)^2}{\sigma_{WR}^2} < \theta_{as}^2 \quad (3)$$

- $M_{as}$  = mittlere, skalierte Bioäquivalenz
- $\mu_T$  = Mittelwert der Testformulierung
- $\mu_R$  = Mittelwert der Referenzformulierung
- $\sigma_{WR}^2$  = intraindividuelle Varianz der Referenzformulierung
- $\theta_{as}^2$  =  $\theta_A / \sigma_{w0} = \log(1.25)^2 / 0.25^2 \sim 0.7968871$

Im Unterschied zur ABE-Auswertung müssen hierbei jedoch Studien mit wiederholter Gabe der gleichen Formulierung (replicate design) durchgeführt werden, um  $\sigma_{WR}^2$  bestimmen zu können.

Abbildung 1 verdeutlicht dieses Prinzip. Auf der Abszisse ist die Varianz der Referenzformulierung aufgetragen. Die ABE-Limits sind waagrecht, während sich die SABE-Limits im Nullpunkt treffen. Abhängig von der Grenze  $\sigma_{w0}$ , dem regulatorischen Limit für hochvariable Substanzen, führt die SABE-Auswertung zu einem Aufweiten der Akzeptanzgrenzen.

Ein wichtiges Problem bei der SABE-Auswertung besteht darin, dass die Statistik  $M_{as}$  nicht mehr normalverteilt ist. Zur Berechnung wurden deshalb entweder spezielle Linearisierungen oder Bootstrap-Verfahren vorgeschlagen. Auch wenn beide Verfahren in der Vergangenheit vielfach untersucht worden sind, ist ihre regulatorische Akzeptanz noch unbekannt.



**Abbildung 1:** Die ABE und SABE-Grenzen sowie das 90%-Konfidenzintervall des T/R-Quotienten. Die Standardabweichung der Referenzformulierung beträgt hier 0.34,  $\sigma_{w0}$  wurde mit 0.25 angesetzt.

## 2 Vorgestellte Studie

### 2.1 Hintergrund der Studie

Die Formulierungen des Wirkstoffes, deren Bioäquivalenz in der vorgestellten Studie nachgewiesen werden sollte, besitzen eine hohe pharmakokinetische intra- und interindividuelle Variabilität. In mehreren früheren 2x2 Crossover-Studien wurden für AUC und  $C_{max}$  jeweils intraindividuelle ANOVA-CVs (zwischen Test und Referenz) von 45% ermittelt, so dass die Voraussetzung einer „hochvariablen Substanz“ gegeben ist. Die intraindividuelle Variabilität nach wiederholter Gabe der gleichen Formulierung wurde zuvor noch nicht untersucht.

Auf Grund der vorliegenden Ergebnisse war abzusehen, dass bei Auswertung einer 2x2 Crossover-Studie mit ABE mehr als 140 Probanden notwendig wären, um eine Power von mindestens 90% abzusichern.

### 2.2 Studiendesign

Die vorgestellte Studie wurde daher mit einem replikativen Design mit den Sequenzen RTRT bzw. TRTR geplant. Durch diese gewählte Form ist eine geringere Anzahl an Probanden erforderlich. Die Probanden bekommen von der Referenz- und Testsubstanz

jeweils zwei verblindete Einmalgaben. Die auszuwertenden Pharmakokinetik- Parameter, d.h. die primären Endpunkte, sind AUC und  $C_{\max}$  von zwei Analytes des Wirkstoffes.

Aus der Fallzahlberechnung ergab sich für eine Power von 90% mit SABE-Auswertung eine Fallzahl von 60 Probanden. Mit einer ABE-Auswertung würde diese Fallzahl nur zu einer Power von  $< 65\%$  führen. Um auf mögliche Drop-Outs vorbereitet zu sein, wurden in die Studie 66 Probanden eingeschlossen, die in vier Studienkohorten behandelt wurden.

## 2.3 Analysenplanung

Als primäre Auswertung der Studie wurde SABE verwendet, während ABE zusätzlich als sekundäre Auswertung geplant war, da letztere in jedem Fall für Behörden akzeptabel ist.

Wie in Abschnitt 1.2 ausgeführt, wurde die SABE-Auswertung mit Hilfe von Bootstraps (siehe 3.5) und mit der genannten Linearisierung ausgeführt. In dieser Studie wurde  $\sigma_{w0} = 0.25$  gesetzt, wie in Abbildung 1 dargestellt.

Da die Formel (3) eine Quadrierung der Grenze enthält, wird nur noch einseitig getestet.

Als Sensitivitätsanalyse wurde auch die Average BE mit Hilfe von Bootstraps ausgewertet.

## 3 Programmtechnische Umsetzung

Bei dieser Studie sollte die Auswertung eine Woche nach Erhalt der Daten (berechnete AUC und  $C_{\max}$ ) abgeschlossen sein, wobei eine Vielzahl für Displays zur Darstellung der Ergebnisse geplant war.

Da dieser kurze Zeitraum nicht zur Programmierung der Analyse ausgereicht hätte, konnte die Studiauswertung nur mit Frontloading erfolgreich implementiert werden: Die gesamte Auswertung sollte vollständig programmiert sein, bevor die eigentlichen Studiendaten vorlagen. Für die Programmierung wurde SAS<sup>®</sup> in der Version 8.2 verwendet.

### 3.1 Simulationsstudie zur Generierung der Datenstruktur

Als Datenbasis der Programmierung wurde eine abgeschlossene Studie mit ebenfalls replikativem Design ausgewählt, bei der die Auswertung nach ABE vorlag, jedoch nicht nach SABE. Auch in dieser Studie war die Fallzahl vergleichbar (68 vs. 66 Probanden). Das Studiendesign war jedoch unterschiedlich (z.B. mit den Sequenzen TRRT und RTTR) und musste entsprechend angepasst werden.

Die zuvor erhaltenen ABE-Ergebnisse konnten anschließend zur Validierung der neuen Programmierung genutzt werden.

### 3.2 Analysis Data Sets - ADS

Bei Boehringer Ingelheim werden die Daten in Views aus Oracle\*Clinical abgelegt. Doch findet die Auswertung nicht auf diesen Views statt, sondern es werden für die Analysen sogenannte Analysis Data Sets (ADS) generiert, in denen abgeleitete Analysenvariablen in einem gesonderten permanenten SAS®-Dataset abgespeichert werden. Somit basiert die Auswertung nicht auf Views, sondern auf diesen sogenannten ADS.

Neben dem Vorteil, dass alle relevanten Variablen und Ableitungen, wie sie für die Auswertung gebraucht werden, in einem Dataset zusammengestellt sind, sind keine Datenmanipulationen während der Auswertung und Ergebnisdarstellung mehr notwendig. Stattdessen erfolgt nur eine Datenselektion aus dem ADS. Somit wird die statistische Auswertung und Programmvalidierung vereinfacht. Gleichzeitig werden Qualität und Effizienz der Datenauswertung erhöht.

Ein weiterer Vorteil, der dabei entsteht ist, dass bei einer studienübergreifenden Auswertung innerhalb von Entwicklungsprojekten die Auswertung vereinfacht wird, da alle ADS eine gleiche Grundstruktur haben, bzw. eine Mindestanforderung an Analysenvariablen.

Für pharmakokinetische Studien wurde bei Boehringer Ingelheim ein spezieller ADS definiert, dessen Struktur in der vorliegenden Studie adaptiert wurde. Die Daten der Simulationsstudie wurden in dieses Datenformat übertragen, und alle weiteren Auswerteprogramme wurden auf dem ADS aufgesetzt.

### 3.3 Steuerung der Datenquelle

Während der Entwicklungsphase sollten alle Programme auf das ADS zugreifen, das die simulierten Daten enthält. Nach Erhalt der Daten sollte das Original-Dataset als Datenquelle dienen. Um später kein Vertauschen der Programme zu riskieren, wurden zwei LIBNAME-Statements generiert: eines das den simulierten ADS enthält und eines das den eigentlichen, also den Original- ADS enthält. Da es sehr viele Programme gab, sollte die Datenquelle von zentraler Stelle aus gesteuert werden.

Eine einfache Lösung ist es, eine dynamische Steuerung des Pfades durch Definieren eines LIBNAME-Statements in einer zentralen Datei zu ermöglichen. Dazu werden in dieser zentralen Datei beide LIBNAME-Statements definiert. Eine globale Makrovariable enthält dann den Namen des aktuell gewählten LIBNAME-Statements. In den Programmen erfolgt der Zugriff nur noch über die globale Makrovariable, die einmal programmiert, nicht mehr geändert werden muss:

Zentrale Datei:

```
%GLOBAL inads;  
LIBNAME ads_sim      "C:\KSFE\simu\";  
LIBNAME ads         "C:\KSFE\ads\" ;  
%LET inads = <ads , ads_sim> ;
```

Umsetzung im Programm:

```
DATA gsteuert;
  SET &inads..daten;
RUN;
```

### 3.4 Auswertung auf Makro-Basis

Den vielen darzustellenden Displays lag eine relativ geringe Anzahl von Display Templates zugrunde, da diese Display Templates für die verschiedenen Endpunkte und Auswertungen gleich sein sollten.

Zur Orientierung wurden Standard-Outputs von 2x2 Crossover-Bioäquivalenzstudien verwendet.

Insgesamt sollten 124 Tabellen und Listings sowie 72 Abbildungen erstellt werden, wobei nur 21 Displaystrukturen notwendig waren.

Dazu wurde für jedes Display-Template ein flexibles Makro geschrieben, so dass die Darstellung der Ergebnisse nur noch mit Makroaufrufen erfolgen musste.

### 3.5 Spezielles Verfahren: Bootstrap

Wie bereits erwähnt, kann für die SABE-Metrik  $M_{as}$  keine Normalverteilung angenommen werden. Die Schätzung der Konfidenzintervalle wurde deshalb mit Hilfe eine Bootstrap-Analyse gewonnen.

Beim Bootstrap werden aus einem Original-Dataset der Größe  $N$  neue Datasets gleicher Größe durch Ziehen mit Zurücklegen gezogen. Aus jedem einzelnen Bootstrap-Dataset werden dann die Variabilität der Referenzformulierung ( $\sigma_{WR}^2$ ) und das T/R-Verhältnis der Bioäquivalenzmetrik geschätzt, um  $M_{as}$  entsprechend Formel (3) zu bestimmen. Mit Hilfe der Perzentilmethode wurden anschließend die Verteilungsparameter bestimmt und die SABE-Konfidenzintervalle berechnet.

#### 3.5.1 Erzeugung von zufälligen, aber reproduzierbaren Bootstrap samples

Eine technische Vorgabe für die Generierung der Probandennummern war, dass die erstellten Bootstrap samples zufällig, aber auch reproduzierbar sein sollten. In SAS<sup>®</sup> gibt es unter anderem die Funktion `RANUNI(seed)`, die zur Generierung einer zufälligen Zahlenfolge auf dem Intervall  $[0;1]$  verwendet werden kann. Seed ist bei dieser Funktion der Startwert und liefert bei einem gewählten seed grösser als 0 eine reproduzierbare Zahlenfolge innerhalb eines gleichen Datasteps. Da eine zufällige seed-Nummer in die Funktion eingehen sollte, wurde für jeden der 4 Endpunkte mit Hilfe eines 10-seitigen Würfel eine Zahl gewürfelt. Diese gewürfelte Zahl ist zufällig und die Bootstrap samples durch diese Zahl ( $\geq 0$ ) sind nun reproduzierbar.

### 3.5.2 Generierung aller Probandennummern in einem Datastep

Da 2000 Bootstrap samples generiert werden sollen, die aus je 66 Probanden bestehen, muss zunächst eine Zahlenfolge aus  $2000 * 66$ , d.h. 132.000 Zahlen in einem Datastep generiert werden, um nicht für jedes Bootstrap-Dataset eine eigene Seednummer bestimmen zu müssen:

```
%MACRO boot (datin = , samp = , obse= , seedm = );  
  
    DATA bootges ;  
        RETAIN seed_1 &seedm;  
        DO ptnonew = 1 TO &obse* &samp ;  
            selptno = CEIL (ranuni (seed_1) * total);  
            SET &datin.2 POINT = selptno nobs = total;  
            datas = INT((ptnonew-1)/&obse)+1;  
            OUTPUT;  
        END;  
        STOP;  
    RUN;  
  
    ...  
%MEND;  
%boot (datin = kpkpfas, samp = 2000, obse=66, seedm = 7225449);
```

→Mit der Multiplikation der Anzahl der Probanden (“ \*total “) wird das Intervall auf [1;66] vergrößert. Mit Hilfe der „datas“-Variablen erfolgt in dem großen Dataset ‚bootges‘ die Zuweisung des jeweiligen Bootstrap samples, um das große Dataset anschließend in 2000 kleine Datasets sequentiell unterteilen zu können.

	DATAS	PTNONEW	Patient number	TPATT
58	1	58	59	TRTR
59	1	59	19	TRTR
60	1	60	11	RTRT
61	1	61	28	TRTR
62	1	62	5	TRTR
63	1	63	31	TRTR
64	1	64	66	TRTR
65	1	65	65	RTRT
66	1	66	25	RTRT
67	2	1	16	TRTR
68	2	2	11	RTRT
69	2	3	57	TRTR
70	2	4	28	TRTR
71	2	5	15	TRTR
72	2	6	55	RTRT
73	2	7	29	RTRT
74	2	8	11	RTRT
75	2	9	56	TRTR
76	2	10	60	RTRT
77	2	11	38	TRTR
78	2	12	42	RTRT
79	2	13	57	TRTR

**Abbildung 2:** Auszug aus dem Dataset mit allen 132.000 gezogenen Probanden-nummern: die Variable datas unterteilt das grosse Dataset in kleine Datasets mit je 66 Probanden

### 3.5.3 Splitten des Datasets

Das Unterteilen des Datasets mit den 132.000 Records in 2000 Datasets hat sehr viel Rechenzeit in Anspruch genommen. Zunächst wurde hierbei mit einem WHERE-Statement in einem Datastep gearbeitet, das in einer Do-Loop-Schleife abgearbeitet wurde.

```
%DO j = 1 %TO 2000;
  DATA boot&j;
    SET bootges;
    WHERE datas=&j;
  RUN;
%END;
```

Mit dem Zusammenführen der Originaldatensätze (siehe Abschnitt 3.5.4) und der Berechnung der Teststatistik wurden dabei für jede Zielvariable 16 Stunden Rechenzeit in Anspruch genommen. Für eine Woche Auswertungszeit war das nicht akzeptabel. Der Grund dafür liegt darin, dass im Datastep 2000-mal nacheinander nach der richtigen Kennungsvariablen `datas = &j` in dem riesigen Dataset „bootges“ gesucht wird (Record für Record).

Wegen eines Cache-Problems selektiert SAS® am Anfang der Schleife sehr schnell die Datensätze und wird bei einem höheren Index der Schleifenvariable sehr langsam. Das liegt daran, dass zu Beginn der Schleife der Arbeitsspeicher leer ist. Mit fortschreitender Index-Variablen füllt sich das Cache und es bleiben somit weniger Kapazitäten übrig.

z.B. bei der Indexvariablen = 1 braucht es zum Selektieren 0.01 Sekunden (Abbildung 3) und für die Indexvariable 612 bereits 6.57 Sekunden (Abbildung 4). Beim direkten Selektieren des jeweiligen Datasets, d.h. außerhalb der Schleife, waren alle Zugriffe sehr schnell, d.h auch die für „datas = 1“ und „datas =612“ (0.01 und 0.03 Sekunden), wie in Abbildung 5 zu sehen ist.

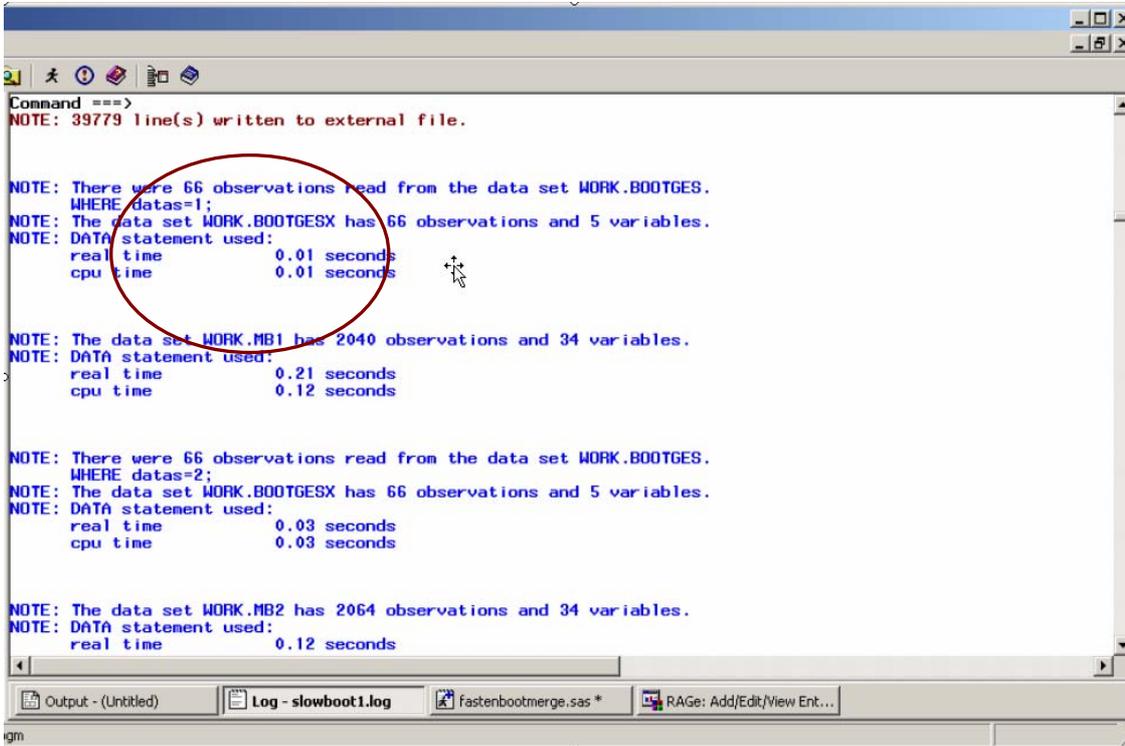


Abbildung 3: Selektieren aus dem großen Dataset bootges: Indexvariable liegt bei 1

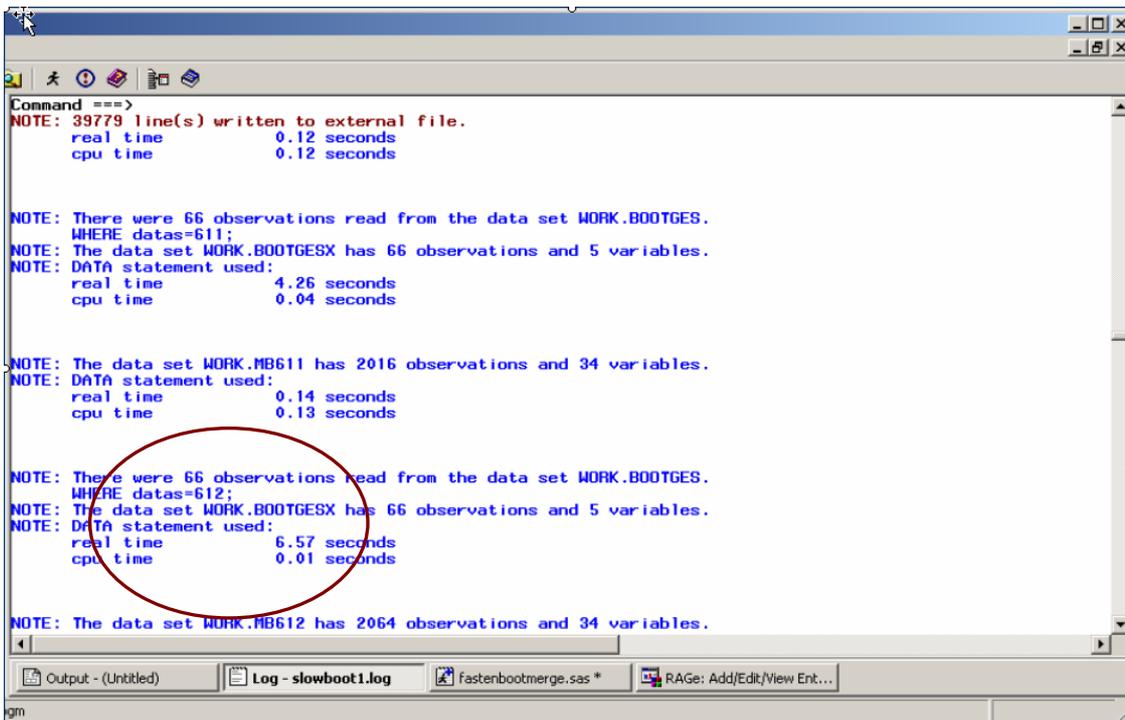


Abbildung 4: Selektieren aus dem großen Dataset bootges: Indexvariable liegt bei 612

```

Command ==>
NOTE: 39779 line(s) written to external file.

3426
3427 data mj2;
3428   set mj;
3429   where datas=1;
3430 run;

NOTE: There were 66 observations read from the data set WORK.MJ.
      WHERE datas=1;
NOTE: The data set WORK.MJ2 has 66 observations and 5 variables.
NOTE: DATA statement used:
      real time      0.01 seconds
      cpu time       0.01 seconds

3431
3432 data mj3;
3433   set mj;
3434   where datas=612;
3435 run;

NOTE: There were 66 observations read from the data set WORK.MJ.
      WHERE datas=612;
NOTE: The data set WORK.MJ3 has 66 observations and 5 variables.
NOTE: DATA statement used:
      real time      0.03 seconds
      cpu time       0.03 seconds

```

**Abbildung 5:** Direktes Selektieren aus dem großen Dataset für die Indexvariablen 1 und 612

Zur Lösung dieses Zeitproblems wurde hier mit PROC SQL gearbeitet.

```

%DO j = 1 %TO 2000;
  PROC SQL;
    CREATE TABLE boot&j AS
      SELECT a.*, b.ptnonew, b.datas
      FROM &datin a, bootges b
      WHERE b.datas = &j AND a.ptno = b.ptno;
  QUIT;
%END;

```

Im Ergebnis konnte damit die Rechenzeit von ca. 16 Stunden auf 30 Minuten pro Endpunkt reduziert werden, obwohl auch hier mit einer Do-Loop-Schleife gearbeitet wird. Der Grund dafür liegt darin, dass das QUIT-Statement die PROC-SQL-Prozedur abschliesst und gleichzeitig den Cache leert.

Eine alternative Lösung wäre es gewesen, einen Index auf die „datas“-Variable zu setzen. Das hat, zeitlich gesehen, den gleichen Effekt wie die Verwendung von PROC SQL zum Splitten der Datasets.

```

PROC SQL;
  CREATE INDEX datas ON bootges(datas);
QUIT;

```

### 3.5.4 MERGE im Vergleich zu PROC SQL

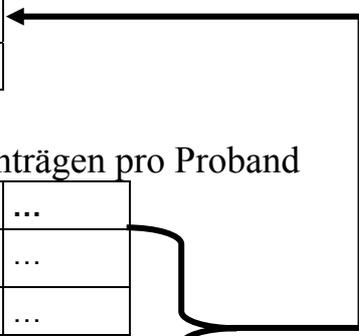
Ein weiterer Grund, warum die Programmierung mit PROC SQL erfolgt, ist folgende: die neu entstandenen Datasets enthalten nun 66 Records mit den neu gezogenen Probandennummern. In die Bootstrap-Datasets sollen aber alle zu den gezogenen Probanden zugehörigen Datensätze mit aufgenommen werden. Jeder Proband hat in der Regel vier Records im Original-Dataset. Dadurch dass, mit Ziehen und Zurücklegen der Probanden, Probandennummern mehrfach vorkommen können, sollen auch Probandeninformationen mehrfach in den Bootstrap-Datensatz eingehen können. Dieser Datenschnitt ist einfacher mit PROC SQL, als in einem Datastep zu realisieren.

**Tabelle 5:** Bootstrap-Dataset mit Probandennummern

seed	ptnew	ptno	tpatt	datas
7725449	1	16	TRTR	1
7725449	2	1	TRTR	1
7725449	3	18	RTRT	1

**Tabelle 6:** Original-Dataset mit mehreren Einträgen pro Proband

ptno	tpatt	Analyt	Endpunkt	y	...
1	TRTR	A1	C <sub>max</sub>	Y1	...
1	TRTR	A2	C <sub>max</sub>	Y2	...
1	TRTR	A1	AUC	Y3	...
1	TRTR	A2	AUC	Y4	...



Folgendes Beispiel soll das verdeutlichen:

Dataset ‚a‘ hat zweimal den Probanden mit der Nummer 1 gezogen. Das Original-Dataset ‚b‘ enthält zu Proband 1 zwei Einträge, die beide doppelt in das Zieldataset aufgenommen werden sollen, da sie auch zweimal gezogen wurden:

```
DATA a;
  ptnew=1; ptno=1; OUTPUT;
  ptnew=2; ptno=1; OUTPUT;
RUN;
```

	PTNEW	PTNO
1	1	1
2	2	1

**Abbildung 6:** Dataset ‚a‘

```
DATA b;
  ptno=1; value = 48.5 ; OUTPUT;
  ptno=1; value = 0 ; OUTPUT;
RUN;
```

	PTNO	VALUE
1	1	48.5
2	1	0

Abbildung 7: Dataset ‚b‘

Während bei einem einfachen MERGE (Dataset ‚d‘) im Datastep nicht die richtigen Datensätze erhalten werden, sind diese im SQL-Dataset (Dataset ‚c‘) wie gewünscht enthalten, da mit PROC SQL ein Kartesisches Produkt generiert wird, wenn man die beiden Datasets über die Variable „ptno“ verknüpft.

```
PROC SQL;
  CREATE TABLE c AS
    SELECT a.ptnnew, b.*
    FROM b, a
    WHERE b.PTNO = a.ptno ;
QUIT;
```

	PTNONEW	PTNO	VALUE
1	1	1	48.5
2	2	1	48.5
3	1	1	0
4	2	1	0

Abbildung 8: Dataset ‚c‘: mit PROC SQL

```
DATA d;
  MERGE a(IN=in1) b(IN=in2);
  BY ptno;
  IF in1 AND in2;
RUN;
```

	PTNONEW	PTNO	VALUE
1	1	1	48.5
2	2	1	0

Abbildung 9: Dataset ‚d‘: mit DATA-STEP und Merge

Um ein Kartesisches Produkt mit einem Datastep - entsprechend dem PROC SQL - zu programmieren, müsste Schritt für Schritt nach den richtigen Datensätzen gesucht werden. Ein Programmierbeispiel ist wie folgt:

```
DATA f;  
  DROP ptnox;  
  DO p1=1 TO nobs1;  
    SET a (RENAME=(ptno=ptnox) )  
        NOBS=nobs1 point=p1;  
  DO p2=1 TO nobs2;  
    SET b NOBS=nobs2 point=p2;  
    IF ptno=ptnox THEN OUTPUT;  
  END;  
END;  
STOP;  
RUN;
```

### 3.5.5 Löschen des temporären WORK-Verzeichnisses

Nach dem Erstellen der 2000 Bootstrap-Datasets werden aus den Mittelwerten des Referenz- und des Testsproduktes die jeweiligen Ratios berechnet. Zusammen mit der Variabilität der Referenzformulierung ( $\sigma_{WR}^2$ ) werden die  $M_{as}$ -Statistiken entsprechend Formel (3) ermittelt und permanent in einem Dataset abgespeichert.

Dadurch können die Konfidenzintervalle mittels Perzentilmethode ermittelt werden.

Das temporäre WORK-Verzeichnis beinhaltet pro Endpunkt 2000 Bootstrap-Datasets und unter anderem weitere 2000 Datasets für die Berechnungen. Diese sollten nach Fertigstellung der Auswertung gelöscht werden, um den Zugriff von weiteren Bootstrap samples erzeugenden Programmen zu verhindern. Beim Löschen des WORK-Pfades mit der PROC DATASETS Prozedur und der Kill-Option stürzte in unserem Fall das Programm ab.

```
PROC DATASETS LIB=work MEMTYPE=DATA NOLIST KILL;  
QUIT;
```

Die Ursache für diesen Programmabsturz konnte nicht genau lokalisiert werden, doch ist ein Cacheproblem als Ursache anzunehmen.

Um dieses Problem zu umgehen, kann mit x-Kommandos gearbeitet werden, die auf Betriebssystemebene Befehle ausführen. Nach Ermitteln des physikalischen Pfades vom WORK-Verzeichnis, kann mit einem DEL-, d.h. DELETE-Statement, der Inhalt des Verzeichnisses gelöscht werden:

```
DATA _null_;  
  LENGTH str2 $500;  
  str2='del ' !! PATHNAME('WORK') !! '\*.sas7bdat';  
  str2=QUOTE(TRIM(str2));  
  CALL SYMPUT('CMD',TRIM(str2));  
  PUT str2=;  
RUN;
```



## 5 Ergebnisse

Wie geplant konnten die Ergebnisse der Analysen etwa eine Woche nach Erhalt der endgültigen Daten in validierter Form präsentiert werden – das Frontloading auf der Basis des simulierten Datasets war damit erfolgreich. Die neuen Auswertemethodiken wurden sehr frühzeitig intensiv getestet, und verschiedenen Probleme, die bei Implementierung auftraten, konnten rechtzeitig erkannt und behoben werden.

Da die Programmierung mit Hilfe flexibel gestalteter Makros erfolgte, könnte diese Auswertung auch auf künftige Bioequivalenz-Studien mit replikativem Design übertragen und eingesetzt werden. Es war darüber hinaus möglich, über das eigentliche Studienziel hinausgehende methodische Untersuchungen effizient durchzuführen.

### Literatur

- [1] Schuirmann DJ. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *J Pharmacokinet Biopharm.* 1987 Dec;15(6):657-80.
- [2] Guidance for industry: *Bioavailability and Bioequivalence Studies for Orally Administered Drug Products - General Considerations (Revision 1)*. U.S. Department of Health and Human Services, Food and Drug Administration, Center for Drug Evaluation and Research (CDER), March 2003.
- [3] Guidance for industry: *Statistical Approaches to Establishing Bioequivalence*. U.S. Department of Health and Human Services, Food and Drug Administration, Center for Drug Evaluation and Research (CDER), January 2001.
- [4] Laszlo Tothfalusi and Laszlo Endrenyi *Limits for the Scaled Average Bioequivalence of Highly Variable Drugs and Drug Products*, Pharmaceutical Research, Vol. 20, No. 3, March 2003