Transnational University Limburg is a unique collaboration of two universities in two countries: the University of Hasselt and Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Application of Bayesian informative priors to leverage historical data in process validation

Gregory Coucke

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Biostatistics

SUPERVISOR :
dr. Jade Vincent MEMBREBE

SUPERVISOR :
Dr. Yimer Wasihun KIFLE

2022
2023

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Application of Bayesian informative priors to leverage historical data in process validation

Gregory Coucke

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Biostatistics

SUPERVISOR :

dr. Jade Vincent MEMBREBE

SUPERVISOR :

Dr. Yimer Wasihun KIFLE

Contents

Abstract

Process validation consists of the collection and evaluation of data, from the process design stage through commercial production, which establishes scientific evidence that a process is capable of consistently delivering quality product. Once the sources of variability are sufficiently identified and controlled during the design phase, a Process Performance Qualification (PPQ) needs to be performed where the production of the product in a validated state is proven on a commercial scale. The sampling plan is of great importance as it defines the intra-batch sample size to be taken during the PPQ to ensure that the chosen process capability metric meets the process specifications with sufficient certainty. In this master dissertation the intra-batch sample size is calculated based on a tolerance interval around the process mean of the potency for two Active Pharmaceutical Ingredients (API), such that it will contain 99% of future sample potencies with a confidence of 95%. Bayesian linear mixed models (BLMM) are applied to account for the uncertainty in the calculation of the tolerance interval and the sample size should be large enough to reach a probability of 95% that the tolerance interval will lie within the 95%-105% reference range. Both early and late design stage data are considered and the power prior, the commensurate prior and a mixture prior are applied to the univariate API data to allow partial borrowing on the residual variance based on the similarity between these data sets. When assigning weakly informative hyperpriors, these methods are shown to lead to the same intra-batch sample sizes. However, when changing the hyperpriors, the sample size based on the commensurate prior seems to better reflect the similarity of early and late design stage residual variance estimates. The power prior and commensurate prior were applied on bivariate API data as well and revealed that only a small increase in sample size is needed to ensure that the tolerance intervals of both outcomes simultaneously lie within the reference range.

1 Introduction

1.1 Process validation

Within a company that produces drug products of human or animal origin, process validation is mandatory (and legally enforceable) to ensure the products are produced with a high degree of assurance of meeting all the attributes they are intended to possess [1]. Process validation is described by the Food and Drug Administration (FDA) as the collection and evaluation of data, from the process design stage through commercial production, which establishes scientific evidence that a process is capable of consistently delivering quality product [1].

Process validation consists of three phases that help to ensure drug quality: process design, process qualification and continued process verification. Through these phases, a high degree of assurance has to be obtained that the process will consistently produce a drug product which meets those attributes relating to identity, strength, quality, purity and potency. In addition, they are meant to understand the sources of variation and the impact they have on the process and the product attributes.

During the process design, the commercial manufacturing process is defined based on knowledge gained through development and scale-up activities. At the laboratory scale, Design of Experiment studies can reveal relationships between process parameters or component characteristics (related to the raw material) and in-process material, intermediates or the final product. This in turn can help to establish ranges for component quality, equipment parameters and material quality attributes. Furthermore, the functionality and limitations of the manufacturing equipment is evaluated during this stage, as well as the variability introduced by different component lots, production operators, environmental conditions, and measurement systems in the production setting.

The process knowledge gained during the process design is used to establish a strategy for process control. Process control tries to maintain the quality of the product by monitoring the process and addressing the observed variability. FDA expects controls to include both examination of material quality and equipment monitoring.

The process qualification stage starts with the design of the facility (if applicable) and the qualification of utilities and equipment. After, the PPQ is performed during which the trained personnel uses the qualified equipment to complete the manufacturing process according to the appropriate control procedures. The PPQ is an important milestone in the product life-cycle and successful completion is necessary to commence commercial production. Data is gathered from the PPQ batches to demonstrate the process is performing as intended on a commercial scale. This master dissertation aims at providing insight into Bayesian methodology that can help to successfully complete the PPQ stage.

It is advised by the FDA that data from all relevant studies (i.e. designed experiments, laboratory, pilot and commercial batches) should be used to establish the manufacturing

conditions of the PPQ [1]. Wherever possible, objective measures (e.g., statistical metrics) should be used to achieve adequate assurance regarding product quality attributes. Within the PPQ protocol the manufacturing conditions, controls, testing and expected outcomes need to be described before starting the execution of the PPQ runs. The sampling plan is herein important to define the number of samples to be taken to provide sufficient statistical confidence of quality both within and between batches. The number of samples to take however impacts the amount of raw material to start with, the number of operators to work on the PPQ and ultimately the length of the study. It is therefore important to statistically motivate a sample size to regulatory authorities that assures uniform product quality throughout the process, yet doesn't cause unnecessary expenses. The Bayesian methodology presented here should assist in determining a sample size that considers both aspects based on both early and late design stage data.

Continued process verification is the final stage of process validation and ensures that the process remains in a continued state of control (the validated state) during commercial manufacture. To accomplish this, product and process data are continuously gathered, statistically trended and reviewed. This allows to detect significant sources of variability and establish appropriate detection, control and/or mitigation strategies. It might provide ways to improve the process through the change of operating conditions, process controls or in-process material characteristics.

1.2 Process capability analysis

Process capability analysis plays a central role in the Bayesian statistics inspired sample size calculation that is performed in this master dissertation. It refers to methods designed to estimate the capability of a manufacturing process to meet a set of requirements or specification limits [2]. In the simplest case, samples are taken and classified as conforming or nonconforming based on whether or not they meet the specifications for the item. The proportion of nonconforming items can easily be calculated and used as a process capability metric.

Sample sizes required to obtain a precise estimate of the proportion of nonconforming items are usually quite high as information is lost on how close or far samples are from the specification limits. If the specifications for a product are based on a continuous variable X, precise estimates may be obtained from much smaller sample sizes by first modelling the probability distribution of X and then using the mean μ and standard deviation σ to determine the capability index. One such measure is the Z index, defined as:

$$Z_{lower} = \frac{\mu - LSL}{\sigma}$$

for the lower specification limit (LSL) and:

$$Z_{upper} = \frac{USL - \mu}{\sigma}$$

for the upper specification limit (USL). The distance to the nearer specification limit is the smaller of the two one-sided Z indices, after which the probability of being beyond the specification limits can be calculated using the standard normal distribution. Another index commonly used is the C_p index:

$$C_p = \frac{USL - LSL}{6 * \sigma}$$

This measure calculates how much wider the "design tolerance" is relative to the "natural tolerance". Statistical tolerance limits on the other hand make a statement about a given proportion of the population at a specified level of confidence and is the method of choice used here. Such an interval can be calculated to define the range within which 99% of future samples would lie with 95% confidence. If this interval lies entirely within the specification limits, it can then be stated with 95% confidence that 99% of the future samples taken will satisfy the specifications. As mentioned earlier, to obtain a precise estimate of process capability, a sufficiently large, representative sample will have to be drawn. How large will depend on the capability index and whether one-sided or two-sided specification limits are used. Here, the sample size is calculated to obtain a 99% coverage 95% confidence (99%/95%) tolerance interval that lies within the specification limit with a probability of at least 95%.

The calculation of the capability index can depend on both information obtained from the samples as well as prior information on the parameters used in the calculation. This is encouraged by the FDA, who state that laboratory or pilot-scale models designed to be representative of the commercial process can be used to estimate the variability. Several informative priors are used here to draw posterior inference on the within batch residual variance using early process design stage data, followed by a risk based calculation of a sample size which is appropriate for PPQ and is based on the use of tolerance intervals.

1.3 Using informative priors for a PPQ sampling plan

Central to Bayesian statistics is the ability to allow inference of model parameters to be influenced by both the data likelihood and prior information:

$$p(\theta \mid D) = \frac{L(\theta \mid D)\pi(\theta)}{\int L(\theta \mid D)\pi(\theta)d\theta} \propto L(\theta \mid D)\pi(\theta)$$

where $\pi(\theta)$ represents the prior on θ and $L(\theta \mid D)$ is the likelihood depending on the data D. The prior can be vague when it is locally uniform on the interval where the likelihood

is not (close to) zero [3]. When such a prior is used, usually posterior summary measures are obtained that are close to those obtained under the frequentist approach. Priors which contain information on the model parameters, as is the case here, are called informative priors. They express skepticism or optimism and can be based on historical data or represent expert opinions. In the next sections the informative priors used in this master dissertation are briefly discussed, together with some of the practical applications they have been used for. Several priors need to be evaluated to demonstrate that the sample size is robust to the prior assumptions.

1.3.1 The power prior

The power prior is one of the most well known informative priors used in Bayesian statistics. It allows discounting of the prior information through the use of the discounting parameter a_0 [4]. The power prior with fixed discounting parameter is defined as:

$$\pi\left(\boldsymbol{\theta} \mid D_0, a_0\right) \equiv \frac{L\left(\boldsymbol{\theta} \mid D_0\right)^{a_0} \pi(\boldsymbol{\theta})}{\int_{\Theta} L\left(\boldsymbol{\theta} \mid D_0\right)^{a_0} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}}$$

When this fixed discounting parameter is given the value 0, the historical data is ignored, while it is given the same weight as the current data when it has the value of one. In the latter case, this also implies that both the historical and current data were generated under identical conditions. As a_0 approaches zero, the tails of the prior become heavier and more uncertainty is introduced in the prior. A hyperprior can be put on the discounting parameter to allow the data to determine the amount of borrowing that should be done. This is then called the joint power prior or unnormalised power prior. Another variation of the power prior where a_0 is random, is the normalised power prior, which first specifies a conditional prior for θ given a_0 and then defines a marginal distribution for a_0. Applications of the power prior are widespread. They have been used in human genetics research to study heritability estimates using twin data [5]. Another example is in the evaluation of the water quality where historical data are used to overcome the inadequate sample size to obtain precise parameter estimates [6].

1.3.2 The commensurate power prior

Combining the normalized conditional power with a prior distribution on the discounting parameter leads to the modified power prior (MPP). While this MPP allows the data to determine the value of a_0, the full conditional posterior for a_0 is free of the current data and as such is not based on a direct comparison between current and historical data. The commensurate power prior does allow this and is different from the MPP in that different parameters are allowed for the historical and current data [7]:

$$\pi\left(\theta, \theta_0, a_0, \tau \mid D_0\right) \propto \frac{L\left(\theta_0 \mid D_0\right)^{a_0}}{\int L\left(\theta_0 \mid D_0\right)^{a_0} d\theta_0} \pi\left(\theta \mid \theta_0, \tau\right) \pi\left(a_0 \mid \tau\right) \pi_0(\tau)$$

Here θ and θ_0 represent one-dimensional parameters based on the current and historical data, respectively. The prior on θ depends on θ_0 and has precision τ, where τ parameterizes the commensurability or agreement between θ and θ_0. In addition the information on τ is used to guide the prior on a_0. The latter is usually a beta prior where a_0 depends on a function which increases with τ. Therefore, as τ approaches 0, the conditional prior variance of θ increases and the borrowing power will decrease through the prior on a_0. For Gaussian data, both power and commensurability parameters inflate the conditional prior variance of θ given weak evidence for commensurability. When only a single historical study is used, a commensurate prior is instead recommended to weight the influence of prior information:

$$\pi\left(\theta \mid D_0, \theta_0, \tau\right) \propto L\left(\theta_0 \mid D_0\right) \pi\left(\theta \mid \theta_0, \tau\right) \pi_0(\theta)$$

The historical data will be ignored when τ approaches zero as this will lead $\pi\left(\theta \mid D_0, \theta_0, \tau\right)$ to approach $\pi_0(\theta)$.

1.3.3 The robust mixture prior

Special attention is given in this master dissertation to the use of priors that allow dynamic borrowing based on the similarity of the residual variance distribution of the current and historical data. This approach should allow to justify to legal authorities the within batch sample size that is taken during the PPQ while including early design stage data. Another approach that allows dynamic borrowing is the robust mixture prior.

The robust mixture prior is defined by considering an informative prior, based on the historical data (M_{inf}), and a vague prior (M_{vag}) together with prior weights that express the prior belief that the historical and current data are similar [8]:

$$\pi(\theta) = p(M_{inf})\pi\left(\theta \mid M_{inf}\right) + p(M_{vag})\pi\left(\theta \mid M_{vag}\right)$$

After considering the current data, the conditional posteriors under each model are updated separately and the prior model probabilities are updated to obtain the posterior model probabilities. The posterior distribution is then a weighted average of the posterior distributions under each model, weighted by their respective posterior model probabilities. This approach was performed to show efficacy of mepolizumab in adolescent patients with severe asthma, using the results in adults to construct a mixture prior [8]. Mixtures of prior distributions have also been used to perform predictive Bayesian sample size calculations in clinical trials [9].

1.4 Bayesian sample size calculation using historical data

Estimating the sample size is important for any experimental design to gain sufficient power to find a significant effect size. In clinical trials, it is mandatory for both budgetary and ethical reasons to avoid exposing too many patients to the experimental treatment arm. Next to the classical frequentist approach, there exist hybrid classical and Bayesian as well as fully Bayesian methods to estimate the required sample size [10].

A disadvantage of the classical approach is that usually an initial guess needs to be provided for a parameter that controls the sampling distribution of the statistic needed for inference. Bayesian methods don't suffer from this local optimality problem as they allow to model the uncertainty through prior distributions [11]. The sample size can be determined using the Bayesian conditional or unconditional power function or can be based on other posterior measures. A hybrid classical and Bayesian method is usually preferred over a fully Bayesian approach in a clinical setting. Here, a distinction is often made between a design prior used to determine the prior predictive distribution and an analysis prior incorporated to perform posterior inference [12]. The design prior can be informative while the analysis prior is usually kept vague. The power prior as well as a mixture of informative priors have been used for sample size calculations [11] [9].

In this master dissertation, simulated early and late design stage data are considered from a process for which the intra-batch sample size has to be determined to successfully complete the PPQ. This is necessary to show the process is able to consistently produce product on a commercial scale with quality attributes that lie within the reference range.

Partial borrowing is performed on the residual variance using informative priors to ensure the sample size is large enough to achieve a probability of 95% that the 99%/95% tolerance interval around the late design stage process mean will lie within the reference range of the process. Borrowing for the process mean is not appropriate as several critical process parameters, necessary to ensure the desired product quality, are usually still prone to change at the early stage. BLMM's are fitted to the historical data and then used as informative priors after being appropriately discounted using either the joint power prior, the commensurate prior and a robust mixture prior. This is done first for both API outcomes separately. After it is assessed whether these methods can be applied to both potency outcomes when they are modelled jointly. Several priors are evaluated to achieve a robust sample size that can be sufficiently motivated to legal authorities and allows a cost efficient planning for the company.

2 Data description

The data for this thesis project was provided by Yimer Wasihun Kifle who works as a Senior Statistician at Janssen, Pharmaceutical companies of Johnson & Johnson. For legal reasons, the data had to be simulated. It consists of potency measurements of API of samples taken throughout the early and late design stage of the process. The API potency is an example of a critical quality attribute, which needs to be controlled throughout the process to maintain the quality of the product. It is expressed as the percentage of the drug's label claim and is here required to be in a 95% to 105% reference range.

The data from the early and late design stage are referred to throughout this dissertation as the historical and current data respectively. In addition, a potency measurement on two API's was generated for each sample: API1 and API2. These are two different ingredients that are both part of the same combination drug. The current and historical data differ primarily in the number of samples per batch, as well as the amount of batches the samples were taken from. These differences are detailed in Table 1.

Table 1: Design stage data summary

Data set	Samples per batch	Batches	Total	API1 mean (SD)	API2 mean (SD)
Historical	20	20	400	99.45 (1.49)	97.84 (1.23)
Current	5	10	50	100.42 (1.37)	97.99 (0.95)

The potency measurements that were performed on the dosage units from the historical and current batches are represented in Figure 1. The overall process mean is higher for API1 than API2 for both the current and historical data. The variability is noticeably higher for the historical data than it is for the current data. For the historical batches the process is still evaluated to determine the parameters that contribute the most to the variability that is introduced in the process. The final reference ranges might also not have been set yet in order to control this variability more efficiently. The correlation for both potency outcomes is very high, with a Pearson correlation of 0.992 for the current data and 0.985 for the historical data. While all the potency values for the current batches are within the reference range, several historical batch units have crossed the lower 95% potency limit. This is true for API1 (one dosage unit for batch 7) and API2 (four units for batch 4 and three for batch 7).

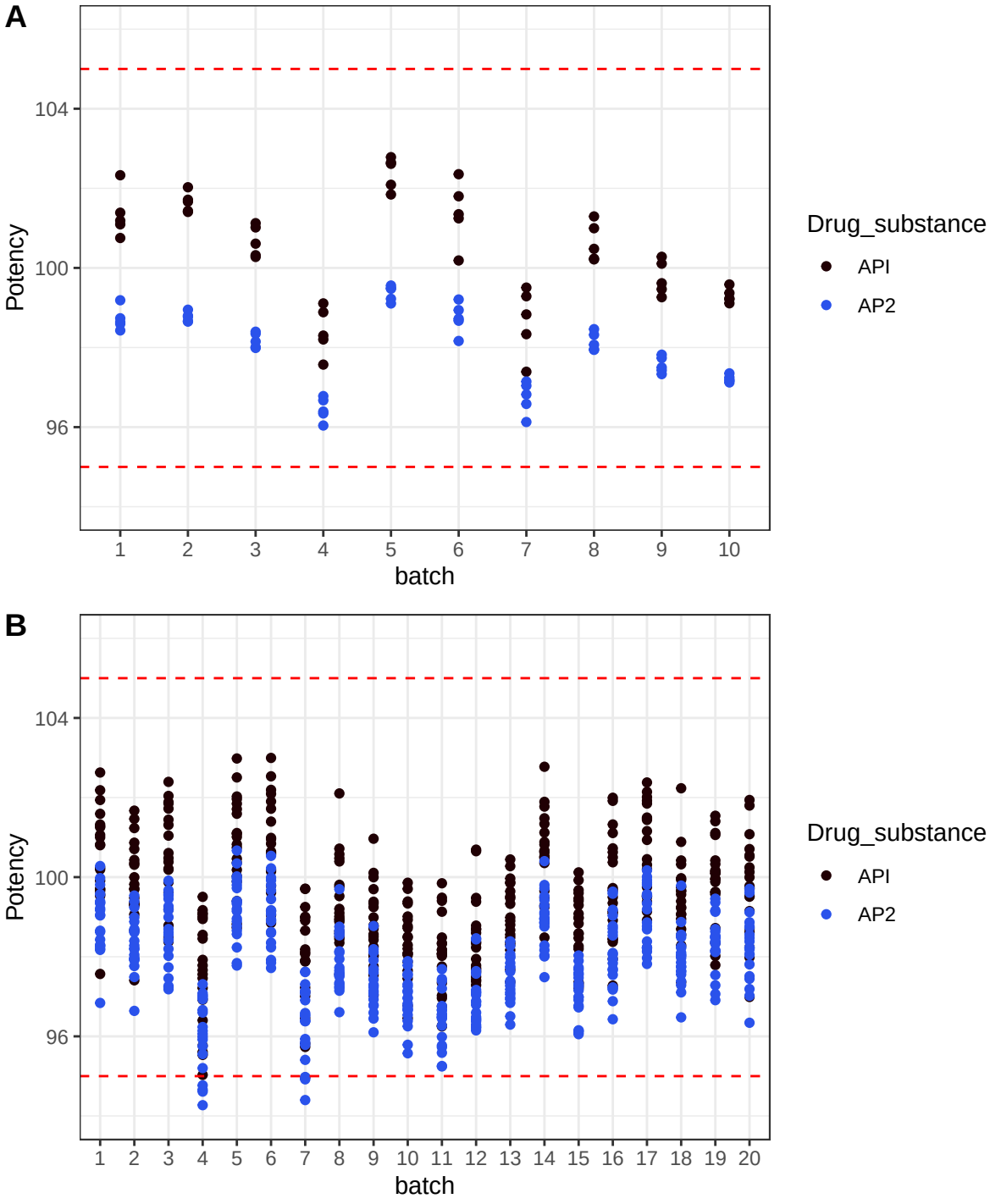

Figure 1: Potency for dosage units of current (A) and historical (B) batches.

3 Methods

In this master dissertation, the intra-batch sample size required for PPQ is determined through a risk based analysis to allow the 99%/95% tolerance interval of the overall process mean to lie within the API potency limits with a probability of at least 95%. BLMM's are used to explore the marginal posterior of the intra-batch variance which allows to account for the variability of the residual variance estimate in the sample size calculation. Sampling of the posterior is performed using the No-U-Turn Hamiltonian Monte Carlo sampling (NUTS) algorithm that can be applied through the Stan software. Assessing model convergence is done through traditional methods such as the trace plot, as well as additional diagnostics that are retrieved when performing NUTS sampling. The influence of the historical data on the late stage intra-batch variability is controlled through several informative priors to determine an appropriate sample size for the PPQ stage.

3.1 Posterior sampling using Stan

Hamiltonian Monte Carlo (HMC) is based on the introduction of an auxiliary momentum variable r_d for each model variable θ_d and approaching this augmented model as a fictitious Hamiltonian system where θ represents a particle's position and r_d denotes the momentum of that particle in the d^{th} dimension [13]. The leapfrog integrator is then used to simulate the evolution over time of the Hamiltonian dynamics of this system. For each sample, the momentum variables are first resampled from a standard multivariate normal distribution. After, the leapfrog updates are performed to the position variables θ and momentum variables r to generate a proposal position-momentum pair, which is accepted or rejected through the Metropolis algorithm.

While HMC is less sensitive to correlated parameters and is able to converge much quicker to high-dimensional target distributions, its widespread use is prevented by the need to provide a step size and a desired number of steps. If the step size is too large, the simulation is inaccurate and will yield low acceptance rates. If it is too small, the computation time will quickly increase. While a small number of steps will usually result in slow mixing, defining too many steps can generate trajectories that loop back to retrace previous steps. The Stan software used here relies on the NUTS algorithm which doesn't require these parameters to be fine-tuned yet is shown to perform at least as efficient as HMC [13].

3.2 Convergence diagnostics

Diagnosing convergence of the posterior samples is mandatory to perform reliable inference on the posterior distribution. When performing HMC sampling, a divergence arises when the Hamiltonian trajectory departs from the true trajectory [14]. When the diver-

gence is too high the simulation cannot be trusted as the positions along the simulated trajectory after the Hamiltonian divergence will never be selected as the next draw of the algorithm. As a result, the posterior will not be thoroughly explored. The burn-in and total amount of iterations were chosen to avoid the presence of divergent iterations. In addition four chains were run for each model to reveal multimodality, poor adaptation or mixing [15].

The split-$\hat{R}$ and Effective Sample Size (ESS) are important to respectively determine whether the chains mixed well and the sample size was high enough to obtain a stable estimate of uncertainty. The traditional $\hat{R}$ is calculated by comparing the variance of all chains mixed together to the variance of the individual chains. It is also called the potential scale reduction factor to denote the factor by which the between-chain variation might decline under future simulations. A value close to one means there is little extra inferential precision to be expected when running the chains longer. Here, convergence is evaluated through the rank-normalized split-$\hat{R}$ (from now one referred to as the $\hat{R}$) which was shown to more reliably assess convergence when the chains have different scale parameters or have especially long tails [15]. Model estimates are reported here only when this $\hat{R}$ is below 1.05. Similarly calculated on the rank-normalized draws, the bulk- and tail-effective samples sizes (bulk-ESS and tail-ESS) are retrieved as an overall and tail-specific efficiency measure respectively. They are suggested to be at least one hundred times the number of chains that were specified [15].

3.3 Univariate model and prior specification

The effect of the intra-batch variability on the sample size calculation is first determined for the current and historical batches separately. After, it is evaluated how the current residual variance and associated sample size changes when then historical data is incorporated into the prior specification. This is done through the joint power prior with random discounting parameter, the commensurate prior and the mixture prior containing both an informative and a vague component.

Initially, the brms package in R was used to define the BLMM's for the current and historical data. Once these models were created, the associated Stan code could be retrieved and adapted into the different informative priors. Consequent analysis were performed by executing the Stan code through the Rstan interface package. The evaluation of the model diagnostics and retrieval of the relevant model parameters were all done through functions provided by the Rstan package.

3.3.1 The Bayesian linear mixed model

The model specification under the Bayesian framework is based on both the likelihood and the prior. In the next sections attention is given to the formulation of both to clarify the partial borrowing of historical data and to indicate the difference in parameterization between the informative priors. In addition the prior for every model parameter is specified. To determine the intra- and inter-batch variability of the API1 and API2 potency for the current and historical batches, a BLMM was fitted which has a Gaussian distribution at each hierarchical level and makes the following distributional assumptions [3]:

- Level 1: $y_{ij}|\beta_0, b_i, \sigma^2 \sim N(\beta_0 + b_i, \sigma^2)$ for j=1,...,m_i; i=1,...,n
- Level 2: $b_i|\sigma_b^2 \sim N(0, \sigma_b^2)$ for i=1,...,n
- Priors: $\sigma^2 \sim \pi(\sigma^2)$ and $(\beta_0, \sigma_b^2) \sim \pi(\beta_0, \sigma_b^2)$

where i represents the batch number and j a sample taken from batch i. The parameters β_0 and σ_b^2 are considered independent, such that $\pi(\beta_0, \sigma_b^2) = \pi(\beta_0)\pi(\sigma_b^2)$. Here, y_{ij} is the potency of sample j taken from batch i, b_i is the random batch intercept for batch i, β_0 is the process mean, σ^2 is the intra-batch variance and σ_b^2 is the inter-batch variability. The priors that were specified for each of the model parameters were:

- $\pi(\beta_0) \sim N(\overline{Y_{..}}, 2.5)$
- $\pi(\sigma) \sim t(3, 0, 2.5)$
- $\pi(\sigma_b) \sim t(3, 0, 2.5)$

For each analysis, the average potency was used as the mean of the normal prior that was specified for β_0. A standard deviation (SD) of 2.5 was chosen to keep the prior weakly informative to avoid that unrealistically large values would be sampled that could lead to slower convergence. A half-t distribution was assigned to both the intra- and inter-batch SD, as it is useful when the SD has to be restricted away from very large values but also because it has better behaviour near 0 than for example an inverse-gamma distribution [16]. Based on the likelihood and the priors, the posterior distribution can be defined as:

$$p(\beta_0, \boldsymbol{b}, \sigma^2, \sigma_b^2|\boldsymbol{y}) = \prod_{i=1}^{n}\prod_{j=1}^{m_i} N(y_{ij}|\beta_0, b_i, \sigma^2) \prod_{i=1}^{n} N(b_i|\sigma_b^2)\pi(\sigma^2)\pi(\beta_0)\pi(\sigma_b^2)$$

3.3.2 The joint power prior

Partial borrowing from the historical data based on the power prior is achieved through the shared residual variance σ^2 and is driven by the similarity between the current and historical data. The strength of the borrowing is directed through the discounting parameter a_0 and the prior that is assigned to it. The discounting parameter can take values from 0 to 1 where values closer to 0 will increase the variance of the prior based on the historical data, thus increasing the dependence of the marginal posterior of σ^2 on the current data. To simplify the notation, parameters specific to the historical model will be denoted as θ_0, those specific to the current model as θ_1, when common to both models as θ_c and as $\boldsymbol{\theta}$ to refer to all parameters of the current model. The historical data is defined as D_0 and the current data as D_1. The power prior used here is called the joint power prior and is in general defined as [4]:

$$\pi\left(\boldsymbol{\theta}, a_0 \mid D_0\right) \propto \pi^*\left(\boldsymbol{\theta}, a_0 \mid D_0\right) \equiv L\left(\boldsymbol{\theta} \mid D_0\right)^{a_0} \pi(\boldsymbol{\theta}) \pi\left(a_0\right)$$

Contrary to the modified power prior, the normalising constant is not calculated here. In case of partial borrowing, the unnormalised power prior is defined as:

$$\pi\left(\boldsymbol{\theta}, a_0 \mid D_0\right) \propto \left\{\int L\left(\boldsymbol{\theta}_c, \boldsymbol{\theta}_0 \mid D_0\right) d\boldsymbol{\theta}_0\right\}^{a_0} \pi\left(\boldsymbol{\theta}_c, \boldsymbol{\theta}_0\right) \pi\left(\boldsymbol{\theta}_1\right) \pi(a_0)$$

where $\pi(\boldsymbol{\theta_1})$ defines the priors for parameters specific for the current data and $\pi(a_0)$ represents the prior on the discounting parameter, which is here a beta(alpha,beta) distribution. The priors for β_0,σ and σ_b for the historical and current model are the same as those defined for the BLMM in section 3.3.1. The influence of the parameterization of the beta distribution on a_0 is assessed by changing the alpha parameter from 1 to 10 while keeping beta at 1. Finally a model was evaluated where a gamma(3,2) hyperprior was assigned to both alpha and beta.

3.3.3 The commensurate prior

As apposed to the partial borrowing joint power prior, in the commensurate prior none of the parameters are shared between the historical and current data. Instead, borrowing happens by assigning a prior to σ_1 with mean σ_0 and SD τ. This is represented in the following notation, where $\boldsymbol{\theta_0}$ and $\boldsymbol{\theta_1}$ now refer to all parameters specific for each model, except for σ_0 and σ_1:

$$\pi\left(\boldsymbol{\theta}, \tau \mid D_0\right) \propto \frac{\left\{\int L\left(\sigma_0, \boldsymbol{\theta}_0 \mid D_0\right) \pi(\boldsymbol{\theta}_0, \sigma_0) d\boldsymbol{\theta}_0\right\} \pi\left(\boldsymbol{\theta}_1\right) \pi(\sigma_1) \pi(\tau)}{\int\{\int L\left(\sigma_0, \boldsymbol{\theta}_0 \mid D_0\right) \pi(\boldsymbol{\theta}_0, \sigma_0) d\boldsymbol{\theta}_0\} d\sigma_0}$$

with $\pi(\sigma_1) \propto N(\sigma_0, \tau)$. To evaluate the influence of the prior for τ on the amount of borrowing, several distributions were considered. Both an inv-Gamma(1,1) and an

inv-Gamma(1,0.001) were assigned to τ^2 while a t(3,0,2.5) distribution was specified for τ.

3.3.4 The robust mixture prior

Lastly, dynamic borrowing from the historical data was allowed through a two component mixture prior on σ_1, consisting of an informative and a vague component using λ to indicate the prior belief about the similarity between σ_1 and σ_0. A larger difference in the variability of the current and historical data should be reflected by an increased posterior probability for borrowing from the vague component. This prior is denoted as:

$$\pi\left(\boldsymbol{\theta}, \lambda \mid D_0\right) \propto \frac{\left\{\int L\left(\sigma_0, \boldsymbol{\theta}_0 \mid D_0\right) \pi(\boldsymbol{\theta}_0, \sigma_0) d\boldsymbol{\theta}_0\right\} \pi\left(\boldsymbol{\theta}_1\right) \pi(\sigma_1) \pi(\lambda)}{\int\{\int L\left(\sigma_0, \boldsymbol{\theta}_0 \mid D_0\right) \pi(\boldsymbol{\theta}_0, \sigma_0) d\boldsymbol{\theta}_0\} d\sigma_0}$$

$$\pi(\sigma_1) = \lambda N(\sigma_0, \sigma_{inf}) + (1-\lambda) N(0, \sigma_{vag})$$

A value of 2.5 was taken for σ_{vag} to express the weakly informative character of this component for values close to zero. A first analysis was performed by setting σ_{inf} at the SD obtained for σ_0 when the historical data were analysed separately. The value for λ was hereby changed from 0.5 (allowing the data to determine the amount of borrowing) to 0.9 (prior belief that σ_1 and σ_0 are similar). Secondly, λ was made random and given a beta(alpha,beta) hyperprior, allowing to retrieve the posterior weight associated with the informative component. Finally, λ was given a uniform prior and σ_{inf} was assigned the inv-Gamma(1,0.001) prior.

3.3.5 Model evaluation

To evaluate the model fit and compare models that were fitted using different priors or different prior parameters, the loo package was used to perform leave-one-out cross validation (loo). Both WAIC and loo are estimates of the expected log pointwise predictive density (elpd) [17]:

$$\sum_{i=1}^{n} \int p_t(\tilde{y}_i) log p(\tilde{y}_i|y) d\tilde{y}_i$$

While WAIC does so by calculating the predictive density for each value of the dataset, loo evaluates the predictive density for value i based on the posterior where i was removed from the dataset. The loo calculated by the package is based on Pareto-smoothed importance sampling which was introduced to prevent the importance ratios from having high or infinite variance.

3.3.6 Bayesian sample size calculation based on tolerance intervals

Once the model has been fitted, the data comprising the marginal posterior of σ_1 can be extracted from the Stan object and be used to account for the variability of this

estimate in the risk based sample size calculation. It is based on tolerance intervals and requires that a sample size is selected which ensures that this capability index lies within the reference range with a probability of at least 95%. This probability increases as the intra-batch sample size increases as a tolerance interval is defined as $\overline{y} \pm k * SD$, where k depends on the sample size, the confidence level and the desired proportion. The k-factor was calculated using the tolerance package. The probability of the tolerance interval of the process mean falling within the acceptance range was calculated using the following procedure:

1. Draw a value σ_{1post} from its posterior distribution
2. Sample n_{max} (the highest sample size to be tested) values from a $N(0, \sigma_{1post})$ distribution
3. Calculate the k-factor for each sample size that needs to be evaluated
4. Define the upper and lower tolerance limit based on the mean, SD and k-factor for each sample with size increasing up to n_{max}
5. Add the value for the upper and lower tolerance limit to the mean of the posterior of β_{01} (the process mean for the current batches)
6. Verify whether the tolerance interval for the process mean lies within the 95%-105% reference range
7. Repeat step 2 to 6 for every value of the posterior for σ_1 and calculate the probability to conform to the process specification

While the above method is described for a single process mean, the procedure can be repeated for every possible value between 95 and 105. This allows to visualise how the required sample size to reach the target changes as the process mean ends up closer to the reference limits.

3.4 Bivariate model and prior specification

Performing the sample size calculation based on the probability that the tolerance intervals of both API1 and API2 simultaneously lie within the reference range could possibly provide a higher degree of certainty that the PPQ phase will be successfully completed. In addition because of the high correlation between both outcomes, possibly only a small sample size increase is needed to achieve this. To evaluate this, a bivariate BLMM was fitted to the historical and current data, allowing partial borrowing through the power prior and the commensurate prior. Cmdstanr was used which allows to connect to cmdstan and run the code more quickly due to a lower memory overhead.

3.4.1 The bivariate Bayesian linear mixed model

The Bayesian model was fitted to the joint potency outcomes by considering a bivariate normal distribution at each hierarchical level:

- Level 1: $\boldsymbol{y}_{ij}|\boldsymbol{b}_i, \Sigma \sim N(\boldsymbol{b}_i, \Sigma)$ for j=1,...,m_i; i=1,...,n
- Level 2: $\boldsymbol{b}_i|\boldsymbol{\beta}_0, \Sigma_b \sim N(\boldsymbol{\beta}_0, \Sigma_b)$ for i=1,...,n

where $\boldsymbol{y}_{ij}$ represents the potency of API1 and API2 for unit j, taken from batch i. The sample potencies are distributed normally with the random batch intercept vector for batch i denoted as $\boldsymbol{b}_i$ and covariance matrix Σ. At the second level, the batch potencies are normally distributed with population process mean vector $\boldsymbol{\beta}_0$ and covariance matrix Σ_b. Each of the model parameters were assigned the following priors:

- $\pi(\beta_{01}) \sim N(\overline{Y_{1..}}, 2.5)$
- $\pi(\beta_{02}) \sim N(\overline{Y_{2..}}, 2.5)$
- $\pi(\Sigma) \sim IW(4, I)$
- $\pi(\Sigma_b) \sim IW(4, I)$

Here $\pi(\beta_{01})$ and $\pi(\beta_{02})$ represent the priors for the population process mean for API1 and API2 respectively. Both Σ and Σ_b are assigned an inverse-Wishart distribution as prior with four degrees of freedom and the identity matrix as location parameter. The posterior associated with the above mentioned likelihood and priors is:

$$p(\boldsymbol{\beta}_0, \boldsymbol{b}, \Sigma, \Sigma_b|\boldsymbol{y}) = \prod_{i=1}^{n}\prod_{j=1}^{m_i} N(\boldsymbol{y}_{ij}|\boldsymbol{b}_i, \Sigma) \prod_{i=1}^{n} N(\boldsymbol{b}_i|\boldsymbol{\beta}_0, \Sigma_b)\pi(\Sigma)\pi(\boldsymbol{\beta}_0)\pi(\Sigma_b)$$

3.4.2 Bivariate potency based sample size calculation

Both the power prior and the commensurate prior were used to allow an appropriate amount of borrowing from the historical data. In case of the power prior, the bivariate model for the historical and current data shared the same residual covariance matrix Σ, allowing all other parameters to be distinct. The approach was similar to the univariate case, except that now the BLMM described in section 3.4.1 was used. Again, the discounting parameter a_0 was assigned either a beta(1,1), a beta(10,1) or a beta distribution where alpha and beta were random and given a gamma(3,2) prior. The latter prior was chosen to conform to the domain of these parameters and allow an appropriate prior density at values ranging from zero to ten, while decreasing for values even higher.

Central to the commensurate prior is the definition of a commensurability parameter τ which influences the amount of borrowing based on the agreement between the current and historical data. A direct translation of the approach taken in the univariate model was not feasible and didn't lead to acceptable convergence diagnostics. Instead the covariance matrix for the current model, Σ_1, was given an inv-Wishart prior with scale parameter Σ_0 and degrees of freedom (df) tau. The inverse of tau was then given a uniform prior and τ was assigned a lower bound of four. While a lower bound of two led to similar model estimates, it required twice the amount of iterations leading to worse convergence diagnostics. A possible reason for this is that an inv-Wishart with two df is no longer a proper distribution. It was reasoned that in this parameterisation, tau could act as a commensurability parameter where an increase would result in a higher posterior weight associated to Σ_0. After applying the informative priors to borrow historical information, the marginal posterior values for the variance of API1, API2 as well as the covariance could now be extracted and used to perform the tolerance interval based sample size calculation. The sample size calculation was adapted to let it depend on the probability that both API1 and API2 lie within the reference range:

1. Draw a value for the variance of API1 and API2 and for the covariance from the marginal posterior Σ_{1post}
2. Sample n_{max} (the highest sample size to be tested) values from a bivariate normal with mean zero for API1 and API2 and covariance matrix Σ_{1post}
3. Calculate the k-factor for each sample size that needs to be evaluated
4. Define the upper and lower tolerance limit based on the mean, SD and k-factor for API1 and API2, for each sample with size increasing up to n_{max}
5. Add the value for the upper and lower tolerance limits to the posterior process mean for API1 and API2 and verify whether the tolerance interval lies within the reference range
6. Repeat step 2 to 5 for every value of the posterior for Σ_1 and calculate the probability to conform to the process specification

4 Results

4.1 Early vs. late design stage batch dependent sample size calculation

To evaluate the influence of the intra-batch variability and process mean on the sample size calculation, a BLMM was first fitted to the outcomes of the current and historical data. The results are displayed in Table 2 and Table 3 for the potency of late and early design stage batches respectively. Posterior sampling was performed for 10000 iterations across four chains, discarding the first half as warm-up. No divergent iterations were reported and based on the trace plots and convergence diagnostics good mixing and efficient sampling from the posterior was achieved. The trace plots for the variance and covariance (in case of the bivariate BLMM) are included for most models in the Addendum. It also contains the R and Stan code used to generate the results presented here. The fit was evaluated for each model using the loo information criterion (looic) which equals -2 times the log posterior predictive density obtained through the loo procedure. For the current data, it was 93.9 for API1 and 20.8 for API2. For the historical data, it was 1240.1 for API1 and 931.5 for API2.

Table 2: BLMM for current data

	Mean	SD	$\hat{R}$	bulk-ESS	tail-ESS
API1					
β_{01}	100.42	0.48	1	2901	4543
σ_1	0.56	0.07	1	7792	8655
σ_{b1}	1.48	0.4	1	2950	5137
API2					
β_{01}	97.99	0.35	1	3045	3924
σ_1	0.27	0.03	1	7225	8692
σ_{b1}	1.1	0.3	1	2754	4458

Table 3: BLMM for historical data

	Mean	SD	$\hat{R}$	bulk-ESS	tail-ESS
API1					
β_{00}	99.45	0.25	1	2007	3989
σ_0	1.11	0.04	1	16515	14382
σ_{b0}	1.09	0.2	1	2761	5169
API2					
β_{00}	97.85	0.24	1	1439	2510
σ_0	0.76	0.03	1	9551	12062
σ_{b0}	1.04	0.18	1	1912	4253

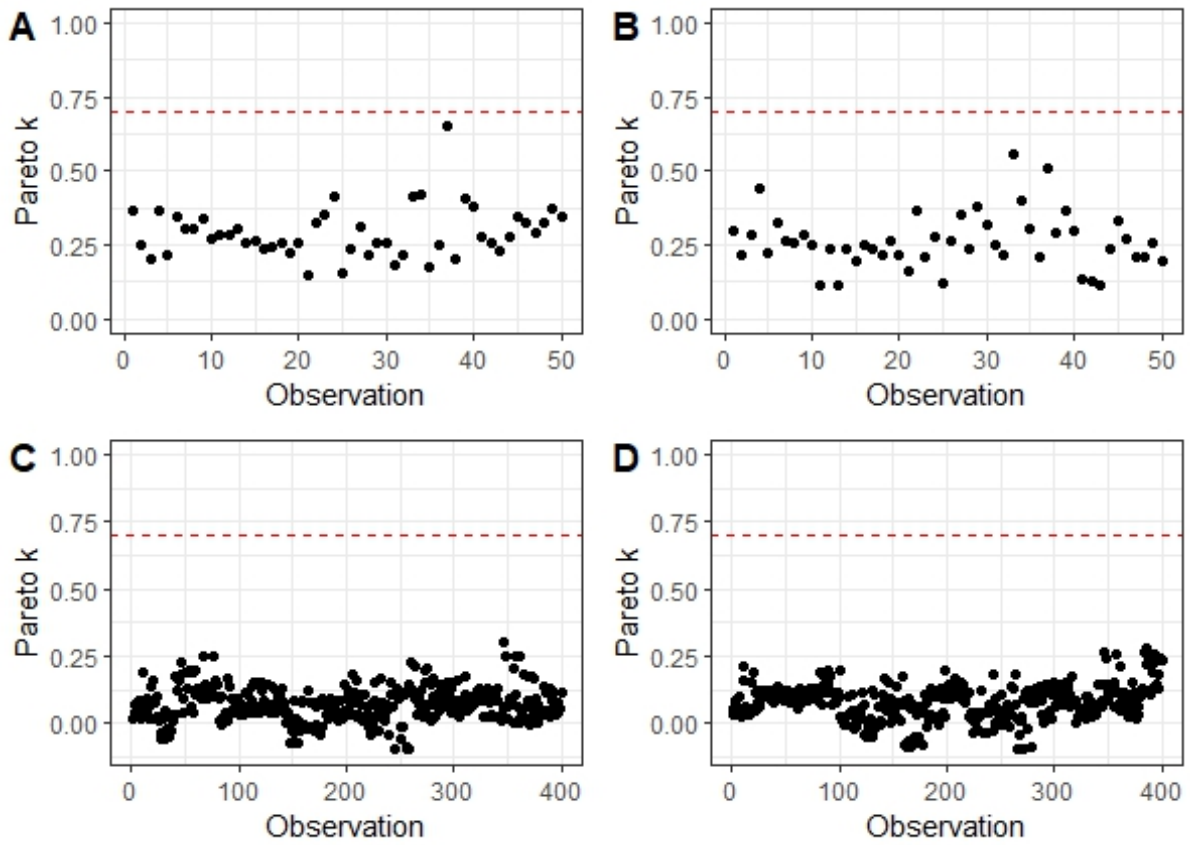

Figure 2: K-diagnostic plot for API1 and API2 of the current batches (A and B), and the historical batches (C and D).

The Pareto k estimates are displayed in Figure 2. Observations can be considered influential when the k-value is higher than 0.7, which points to a significant difference between the posterior with and without this value. No such observations were identified here allowing to conclude a good fit for every model. As expected, the intra-batch variability was higher for the historical batches than for the current batches and was furthermore higher for API1 than for API2. The potency for API1 was closer to 100% than for API2 for both datasets. The relation between the intra-batch sample size and the probability to produce batches within the required reference range is displayed in Figure 3 and Figure 4 for the current and historical data respectively. For the current batches a minimum of 7 samples should be taken based on API1 and 5 samples based on API2. For the historical batches this was 36 and 48 for API1 and API2 respectively. While the intra-batch variability for API2 of historical batches is lower than for API1, the required sample size is a lot higher as the process mean lies closer to the lower acceptance limit.

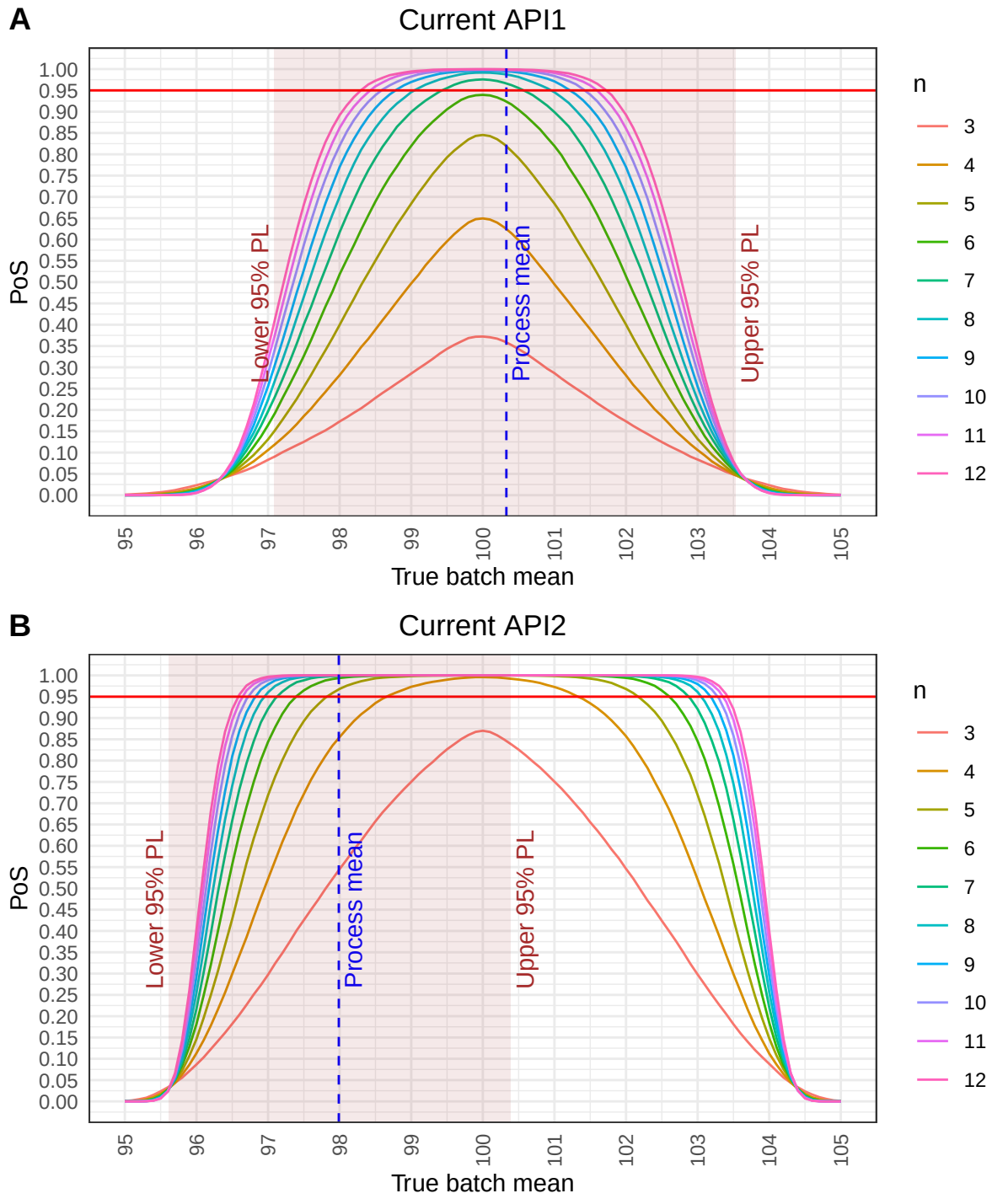

Figure 3: Intra-batch sample size determination for API1 (A) and API2 (B) of the current batches

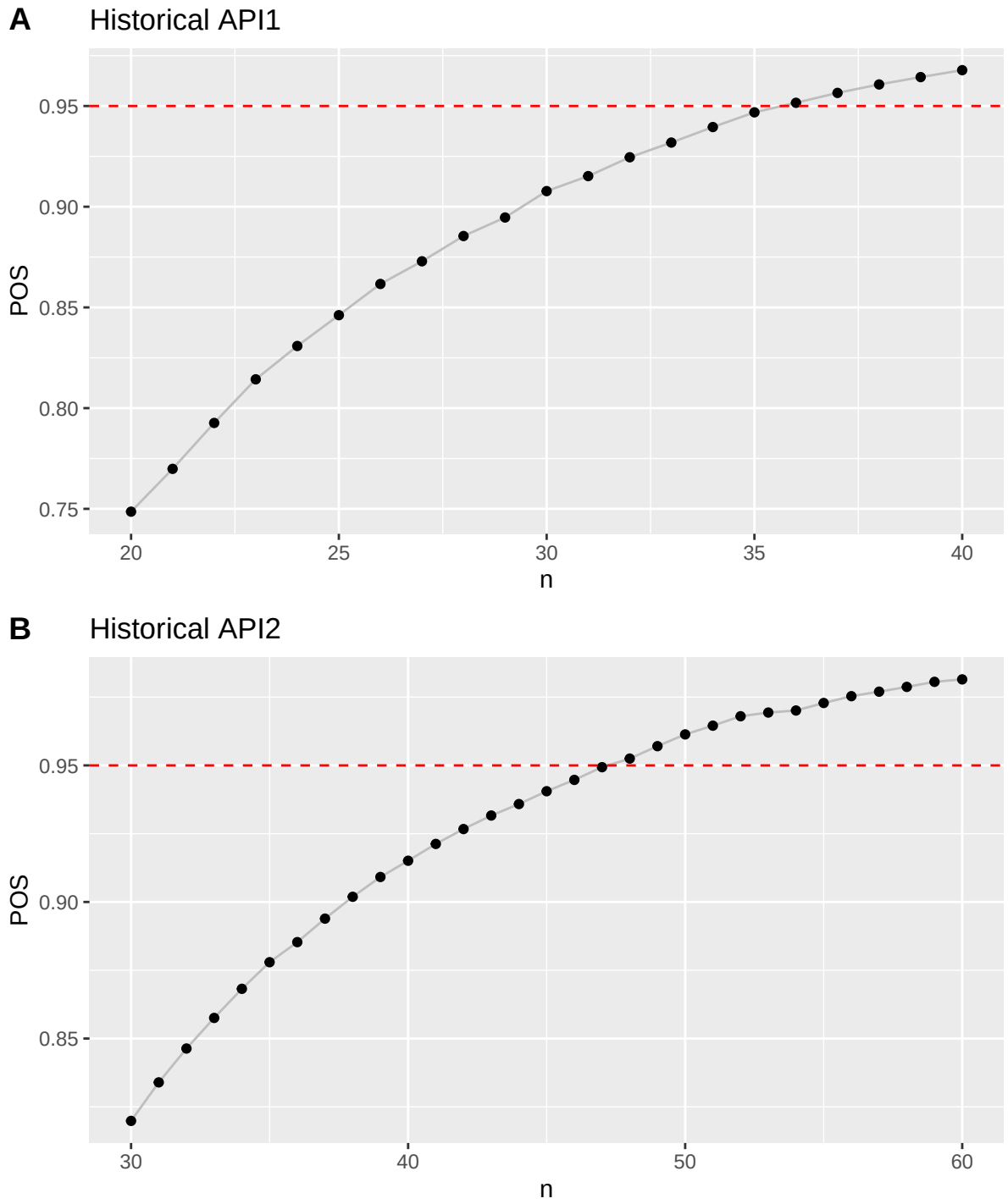

Figure 4: Intra-batch sample size determination for API1 (A) and API2 (B) of the historical batches.

4.2 Borrowing historical data using the power prior

The power prior is one of the most often used informative priors to borrow information from historical studies. The discounting parameter a_0 can be held fixed and a sensitivity analysis can be performed to determine the appropriate amount of discounting that should be performed. Here, a_0 is given a beta(alpha,beta) prior to allow the amount of borrowing to be determined by the similarity of the current and historical data. However, the choice of the hyperprior and the value of the parameters still can influence the amount of borrowing. Therefore the analysis is performed using a beta(1,1) and a beta(10,1) prior as well as by assigning a gamma(3,2) hyperprior to the alpha and beta parameter of the beta distribution.

The model estimates and sample size obtained by assigning different hyperpriors to the discounting parameter are shown for API1 and API2 in Table 4. In addition, the looic is reported. For both API1 and API2, the use of a beta(1,1) hyperprior or the assignment of a gamma(3,2) to the parameters of the beta distribution led to no borrowing and a σ estimate depending solely on the current data. As a result the same sample size was obtained as for the univariate analysis without applying the power prior.

Table 4: Model estimates and sample size using the Power prior

	β_{01}		σ_{b1}		σ		a_0		n	looic
	Mean	SD	Mean	SD	Mean	SD	Mean	SD		
API1										
beta(1,1)	100.42	0.48	1.49	0.42	0.57	0.07	0	0	7	94
beta(10,1)	100.42	0.48	1.46	0.4	0.69	0.09	0.008	0.003	10	97.8
t(3,0,2.5)	100.43	0.47	1.48	0.42	0.56	0.07	0	0	7	93.9
API2										
beta(1,1)	97.99	0.36	1.09	0.3	0.28	0.03	0	0	5	20.9
beta(10,1)	97.99	0.35	1.1	0.32	0.37	0.05	0.0047	0.0021	8	29.8
t(3,0,2.5)	98	0.36	1.1	0.31	0.27	0.03	0	0	5	21.2

For both API outcomes, the assignment of the beta(10,1) distribution resulted in an identical increase of the sample size from 7 to 10 for API1 and from 5 to 8 for API2. This distribution favors higher values of a_0 as it has a density peak closer to one. The mean a_0 for API1 was 0.008 and 0.0047 for API2, which seems to correctly reflect the difference between API1 and API2 in the similarity of σ_0 and σ_1 (obtained when modeling the current and historical data separately). However the mean a_0 for API1 lies close to one SD from that of API2, correctly reflecting the impact of the power prior on σ and the associated sample size calculation. The looic for both models applying the beta(10,1) hyperprior was slightly higher. However, it was not found to be significant based on the pairwise comparison of the pointwise expected predictive density, provided by the compare function of the loo package.

4.3 Commensurability based dynamic borrowing

The parameter of commensurability was introduced here as the SD of the commensurate prior. It is expected to increase as the difference between the current and historical data increases. Depending on the hyperprior put on τ however, the SD can be influenced and the amount of borrowing possibly changed. The parameter estimates of the models and their associated sample size are shown in Table 5. The influence of the use of the vague inv-Gamma(1,1) and t(3,0,2.5) priors on τ is almost identical to the use of the vague hyperpriors for the discounting parameter a_0. The parameter estimates are unchanged and the reported sample size reflects no borrowing from the historical data. This is also reflected in very similar looic values between the power and commensurate prior for these vague hyperpriors. Different is the impact of the inv-Gamma(1,0.001) distribution which favors SD values much closer to zero. While this results in an increase of the sample size from 7 to 9 for API1 it leaves the outcomes for API2 unchanged. Despite encouraging the borrowing from the historical data through this hyperprior, the distribution of σ_1 and σ_0 seems too different to allow borrowing. Again the looic was found to be higher for the inv-Gamma(1,0.001) which was now found to be significant. For both comparisons the mean pairwise difference was -1.4 with SD 0.7.

Table 5: Model estimates and sample size using the Commensurate prior

	β_{01}		σ_{b1}		σ_1		τ		n	looic
	Mean	SD	Mean	SD	Mean	SD	Mean	SD		
API1										
inv-Gam(1,1)	100.43	0.48	1.48	0.48	0.56	0.07	0.95	0.49	7	93.8
inv-Gam(1,10^{-3})	100.41	0.47	1.46	0.4	0.62	0.11	0.3	0.18	9	96.7
t(3,0,2.5)	100.43	0.48	1.48	0.41	0.56	0.07	1.68	1.6	7	93.9
API2										
inv-Gam(1,1)	97.98	0.35	1.09	0.3	0.27	0.03	0.93	0.48	5	20.8
inv-Gam(1,10^{-3})	97.99	0.36	1.1	0.31	0.28	0.03	0.3	0.16	5	20.9
t(3,0,2.5)	97.99	0.36	1.09	0.3	0.27	0.03	1.62	1.63	5	21

4.4 A two component mixture based sample size calculation

The robust mixture prior consists of an informative normal prior centered on the mean of σ_0 and a vague N(0,2.5) distribution. It requires that a value for the SD of the informative prior is chosen. Here it is assigned the SD of σ_0 obtained when no borrowing prior was used. In addition it was given the value 0.3 which is the SD of the commensurate prior when the inv-Gamma(1,0.001) was applied which encouraged borrowing. First the mixing probability λ was kept constant and was defined as 0.5, 0.6 or 0.9 reflecting either an identical prior association with both components or an association heavily favored towards the informative prior. The results are shown in Table 6 for API1 and Table 7 for API2.

Table 6: Application of the mixture prior to API1

	β_{01}		σ_{b1}		σ_1		n	looic
	Mean	SD	Mean	SD	Mean	SD		
API1: SD 0.04								
$\lambda = 0.5$	100.42	0.48	1.48	0.41	0.56	0.06	7	93.7
$\lambda = 0.6$	100.42	0.48	1.48	0.40	0.56	0.07	7	93.9
$\lambda = 0.9$	100.43	0.48	1.48	0.41	0.56	0.06	7	93.9
API1: SD 0.3								
$\lambda = 0.5$	100.44	0.49	1.48	0.40	0.57	0.07	7	94.3
$\lambda = 0.6$	100.45	0.47	1.47	0.4	0.58	0.07	7	94.4
$\lambda = 0.9$	100.42	0.48	1.49	0.41	0.58	0.07	7	93.9

Table 7: Application of the mixture prior to API2

	β_{01}		σ_{b1}		σ_1		n	looic
	Mean	SD	Mean	SD	Mean	SD		
API1: SD 0.03								
$\lambda = 0.5$	97.98	0.36	1.09	0.31	0.27	0.03	5	20.7
$\lambda = 0.6$	98	0.35	1.1	0.31	0.27	0.03	5	20.9
$\lambda = 0.9$	97.98	0.36	1.1	0.31	0.27	0.03	5	20.6
API1: SD 0.3								
$\lambda = 0.5$	97.99	0.36	1.09	0.31	0.27	0.03	5	20.9
$\lambda = 0.6$	97.99	0.36	1.1	0.31	0.27	0.03	5	21
$\lambda = 0.9$	97.98	0.35	1.09	0.31	0.27	0.03	5	21.1

When the SD of σ_0 was assigned as the SD of the informative mixture component (0.04 for API1 and 0.03 for API2) no borrowing was achieved for either outcome and σ_1 stayed constant. In addition, the value for σ_1 did not depend on an increase of λ. However when assigning τ from the commensurate prior after applying an inv-Gamma(1,0.001) hyperprior, a small increase was observed in σ_1 for API1 when λ was increased, but stayed constant for API2. No sample size increase was noted for either API1 or API2. From 0.6 to 1, the prior weight assigned to the informative component is increased until it is ultimately defined to be the only prior distribution.

Finally two additional model specifications were defined for both API1 and API2. First the SD of the informative component was given the value of τ and λ was made random with a uniform hyperprior. Secondly, λ was again assigned a uniform hyperprior while the parameter τ was introduced in the informative component and given the inv-Gamma(1,0.001) hyperprior (identical to the commensurate prior setting). The results of these models are given in Table 8. In both cases, the sample size did not change. The change in the values for τ and λ again demonstrates how borrowing is possibly not appropriate for these datasets. As τ decreases, the mixing probability associated with this distribution decreases to 0.34 while it increases to 0.54 for API1 and 0.56 for API2 for τ equal to 0.3. While the informative component is here centered at σ_0, a SD of

0.3 means that σ_1 is supported with a higher density than when τ is kept random and reported to be 0.04 for both API1 and API2, explaing the decrease of λ.

Table 8: Application of the mixture prior with random λ and τ

	β_{01}		σ_{b1}		σ_1		τ	λ	n	looic
	Mean	SD	Mean	SD	Mean	SD				
Random τ and λ										
API1	100.42	0.48	1.49	0.41	0.56	0.07	0.04	0.34	7	93.9
API2	97.99	0.36	1.1	0.31	0.27	0.03	0.04	0.34	5	20.9
τ=0.3 random λ										
API1	100.43	0.48	1.48	0.41	0.57	0.07	0.3	0.54	7	94
API2	97.99	0.35	1.1	0.31	0.27	0.03	0.3	0.56	5	21.5

4.5 Historical bivariate potency based borrowing

After the application of the informative priors to allow borrowing in a univariate setting, it was attempted to evaluate these methods when both API outcomes were jointly modelled. The function used to calculate the sample size was adjusted to draw residuals from a bivariate normal with mean zero and with scale the covariance matrix of the current data. The process performance was assessed by independently calculating the probability that both API1 and API2 would lie within the reference range, at a Bonferroni corrected confidence level of 0.975. This approach was applied to the current and historical data separately, as well as when allowing historical borrowing using the power prior and a commensurate prior.

The intra-batch variability for each model is detailed in Table 9 and the inter-batch variability is shown in Table 10. The $\hat{\text{R}}$ and ESS met the requirements for all models tested. Some of the models had reports of transitions hitting the maximum treedepth. This doesn't point to biased model estimates but rather highlights an efficiency concern when the algorithm reached the maximum number of simulation steps and cancelled prematurely to avoid excessively long execution time. The maximum treedepth can be adjusted but it was decided not to as this would have excessively increased the computation time.

An increase in the required sample size was observed compared to the univariate setting when the current and historical data were modelled separately. A sample size of 7 was reported for the univariate model based on the late design stage API1 data and now increased to 9 when considering the bivariate potency data. For the historical data, it went from 48 for API2 to 54. When the power prior was applied using a beta(10,1) hyperprior, again an increase was observed to an intra-batch sample size of 10, which equalled the increase of API1 alone when assigning the same prior specification. The looic value was significantly higher compared to using the two vague hyperpriors. When the df of the inv-Wishart prior was defined as the commensurability parameter, no borrowing was achieved and a sample size of 8 was obtained. The value for the df of 6.58 was

consistent with the assignement of a higher posterior weight on the current data based covariance matrix.

Table 9: Intra-batch variability based on bivariate modeling

	σ^2_{API1}		$cov_{API1-API2}$		σ^2_{API2}		n	looic
	Mean	SD	Mean	SD	Mean	SD		
No borrowing								
Current	0.31	0.07	0.14	0.03	0.09	0.02	9	19.7
Historical	1.23	0.09	0.83	0.06	0.57	0.04	54	-231.5
Power Prior								
beta(1,1)	0.31	0.07	0.14	0.03	0.09	0.02	9	21.4
beta(10,1)	0.38	0.09	0.19	0.05	0.13	0.03	10	34.1
t(3,0,2.5)	0.31	0.07	0.14	0.03	0.09	0.02	9	20.1
Commensurate prior								
$\tau = 6.58$ (SD: 1.79)	0.29	0.07	0.15	0.03	0.07	0.02	8	-145.1

Table 10: Inter-batch variability based on bivariate modeling

	$\sigma^2_{b,API1}$		$cov_{b,API1-API2}$		$\sigma^2_{b,API2}$	
	Mean	SD	Mean	SD	Mean	SD
No borrowing						
Current	1.61	0.82	1.09	0.57	0.9	0.45
Historical	1.03	0.36	0.92	0.33	0.93	0.32
Power Prior						
beta(1,1)	1.62	0.83	1.1	0.58	0.9	0.45
beta(10,1)	1.6	0.81	1.08	0.56	0.89	0.44
t(3,0,2.5)	1.63	0.83	1.1	0.58	0.9	0.45
Commensurate prior						
$\tau = 6.58$ (SD: 1.79)	1.63	0.85	1.1	0.59	0.90	0.46

5 Discussion

Process validation is essential to ensure that a quality product can consistently be delivered to the customer that is fit for its intended use. Regulatory authorities such as FDA and EMA oversee through audits that the necessary precautions are in place which allow the production of a drug product that continuous to meet those attributes relating to identity, strength, quality, purity and potency. The successful completion of the PPQ phase is an important milestone in a company and is mandatory to start the commercial release of the product. Determining the amount of samples that should be taken per batch is important to get an accurate process capability estimate and informs the company whether the variability throughout the process is sufficiently controlled to meet all product attribute ranges.

During the design phase of process validation, a wealth of information is gathered on product quality attributes and process parameters. The FDA encourages the use of this information for PPQ. Considering early design stage data in the tolerance interval based sample size calculation could furthermore result in a sample size that more adequately can evaluate whether the range containing 99% of future samples has a 95% confidence to lie within the reference range. Here it was evaluated which informative priors are useful to properly borrow from historical data. When using vague hyperpriors on a_0 for the power prior and τ for the commensurate prior none of the three informative priors seemed to encourage borrowing and the sample size remained unchanged (7 for API1 and 5 for API2). However, earlier it was noted that the power prior tends to overattenuate the impact of the historical data, forcing the use of fairly informative hyperpriors on a_0 to allow sufficient borrowing [7]. In addition, the marginal posterior for a_0 was found to be flat for two identical datasets, regardless of the sample size when a beta(1,1) hyperprior was used. However, here a hyperprior was assigned to both alpha and beta of the beta distribution, leading again to no borrowing.

While the use of vague hyperpriors emphasizes similarities between all three borrowing priors, informative hyperpriors seem to indicate an underlying difference. The difference in variability between the current and historical data seemed greater for API2 than for API1. While the beta(10,1) hyperprior caused borrowing for both API1 and API2, leading to an increased variance and sample size of 10 for API1 and 8 for API2, this wasn't the case for the commensurate prior. The inv-Gamma(1,0.001) prior on τ increased the prior density to values closer to zero, reducing the SD of the commensurate prior for σ_1. While this resulted in an increased intra-batch sample size of 9 for API1, σ_1 as well as the sample size remained unchanged for API2. This possibly points to borrowing that more appropriately considers the similarity between both parameters in case of the commensurate prior, compared to the power prior.

The robust mixture prior is useful when a reasonable estimate of the variance for the parameter of interest is available. Here the variance of the commensurate prior (using the informative inv-Gamma(1,0.001) hyperprior) was taken to demonstrate the dependence of the borrowing on the prior model probability λ. Increasing λ increased the

amount of borrowing from the informative prior and led to an increase of the intra-batch variability for API1 but not API2. The sample size didn't increase. Finally, both λ and the informative component variance were made random thereby resembling a mixture of a commensurate prior and a vague prior. The posterior weight associated with the informative component decreased when its variance decreased, again emphasizing limited data driven borrowing. If an informative hyperprior is indeed required to allow borrowing, the commensurate prior (assigning an inv-Gamma(1,0.001) could be combined with a vague prior in a two component mixture to evaluate whether the similarity between the current and historical data is high enough to warrant borrowing. Additional analyses are required to compare the power prior and the commensurate prior to confirm a possibly higher sensitivity of the commensurate prior to increased parameter disparity. The commensurate power prior is not recommended when only a single historical data set is available. Using several historical data sets, this method could be evaluated as well, in addition to allowing a more appropriate specification (more reliable variance estimate) of the informative distribution of the two component mixture prior.

The application of historical borrowing to the jointly modelled potency outcomes is valuable to further increase the probability to successfully complete the PPQ stage. Its implementation was however challenging and required the use of the inv-Wishart prior for every covariance matrix. The easier sampling of the posterior when using this prior might be due to its conjugacy for a multivariate normal distribution, even though Stan does not require this nor does it require the posterior to be proper. However this observation is in line with the ease with which sampling could be performed using the MCMCglmm package, which depends on block Gibbs sampling in case of conjugacy. Even though models with good convergence diagnostics could be obtained, they required a much longer run time. Performing the sampling using an alternative sampler such as WinBUGS or JAGS could therefore prove valuable.

When the sample size was calculated based on the covariance matrix of the current or historical data separately, only a small increase over the largest univariate sample size based calculation was obtained. Importantly similar conclusions could be made compared to the univariate analysis when the power prior was used. There was no increase in sample size noticeable when a vague prior on a_0 was applied, increasing the variances and covariance only when the beta(10,1) prior was assigned. Due to difficulties in convergence, only a single model could be evaluated that contained the commensurate prior. Considering the degrees of freedom of the inv-Wishart prior as the commensurability parameter seemed intuitive. The df was 6.58, indicating a higher posterior weight put on the current covariance matrix, relative to the historical covariance matrix that was specified as the location parameter of the inv-Wishart distribution. Further research on the application of the commensurate prior would be useful to obtain a more efficient model which would lead to faster convergence. However the approach used here was considered after having already used the brms package and applying the Cholesky decomposition to put a prior on the SD and the Lewandowski-Kurowicka-Joe prior on the correlation matrix, which is also advised in the Stan manual.

While only a small sample size increase was noticed in the bivariate analysis, it could still prove to be an overestimation of the required intra-batch sample size. Both tolerance intervals are considered independently from each other at a Bonferroni corrected confidence level. However the outcomes are highly correlated and a tolerance region should be considered in the multivariate extension of the tolerance interval. This was outside the scope of this master dissertation and requires further attention to evaluate its importance.

6 Ethical thinking, societal relevance, and stakeholder awareness

Quality control of drug products is essential to ensure that a safe product arrives at the customer that is fit for it's intended use. Continued process verification keeps the product under strict quality control when it is commercially produced. However the successful completion of the PPQ is first needed to allow release of the product on the market. If the intra-batch sample size is too low there is a higher risk of not meeting the required process specifications and endangering the commercial release of the product. If it is too high, additional resources will need to be diverted to the PPQ. The methods presented here allow to motivate a sample size to legal authorities that considers both early and late design stage data, properly accounting for the variability in the calculation of the tolerance interval and reducing the risk of choosing a sample size that is too low to stay within the reference range. The application of the partial borrowing informative priors to bivariate data allows a higher reassurance of a successful PPQ stage at the cost of only a small sample size increase. Together it provides a company with the tools to make an informed decision regarding the intra-batch sample size which meets the FDA guidelines of considering all information gathered throughout the design stage of process validation.

7 Conclusion

The power prior and commensurate prior are shown to allow data driven partial borrowing appropriate for the sample size calculation for PPQ in both a univariate and bivariate setting. No sample size increase is obtained when weakly informative hyperpriors are assigned, leading to an intra-batch sample size of 7 and 5 for API1 and API2 modelled separately, and a sample size of 8 (when using the commensurate prior) or 9 (when using the power prior) when the correlation between the API's is taken into account. Further research is required to confirm the higher sensitivity of the commensurate prior to differences between a historical and current study. Finally, the mixture prior could be appropriate to acknowledge the requirement of informative hyperpriors on the one end, but still make a data dependent sample choice on the other.

References

[1] Center for Drug Evaluation and Research. *Process Validation: General Principles and Practices*. Publisher: FDA. 2020.

[2] Neil W. Polhemus. *Process Capability Analysis: Estimating Quality*. 1st edition. Boca Raton: Chapman and Hall/CRC, 2017. 284 pp.

[3] Emmanuel Lesaffre and Andrew B. Lawson. *Bayesian Biostatistics*. 1st edition. Chichester: Wiley, 2012. 534 pp.

[4] Joseph G. Ibrahim et al. "The power prior: theory and applications". In: *Statistics in Medicine* 34.28 (2015), pp. 3724–3749.

[5] Ming-Hui Chen, Amita K. Manatunga, and Christopher J. Williams. "Heritability Estimates from Human Twin Data by Incorporating Historical Prior Information". In: *Biometrics* 54.4 (1998), pp. 1348–1362.

[6] Yuyan Duan, Keying Ye, and Eric P. Smith. "Evaluating water quality using power priors to incorporate historical information". In: *Environmetrics* 17.1 (2006), pp. 95–106.

[7] Brian P. Hobbs et al. "Hierarchical Commensurate and Power Prior Models for Adaptive Incorporation of Historical Information in Clinical Trials". In: *Biometrics* 67.3 (2011), pp. 1047–1056.

[8] Nicky Best et al. "Assessing efficacy in important subgroups in confirmatory trials: An example using Bayesian dynamic borrowing". In: *Pharmaceutical Statistics* 20.3 (2021), pp. 551–562.

[9] Pierpaolo Brutti, Fulvio De Santis, and Stefania Gubbiotti. "Mixtures of prior distributions for predictive Bayesian sample size calculations in clinical trials". In: *Statistics in Medicine* 28.17 (2009), pp. 2185–2201.

[10] Spiegelhalter. *Bayesian Approaches to Clinical Trials and Health-Care Evaluation*. 1st edition. Chichester ; Hoboken, NJ: Wiley, 2004. 407 pp.

[11] Fulvio De Santis. "Using Historical Data for Bayesian Sample Size Determination". In: *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 170.1 (2007), pp. 95–113.

[12] Fulvio de Santis. "Sample Size Determination for Robust Bayesian Analysis". In: *Journal of the American Statistical Association* 101.473 (2006), pp. 278–291.

[13] Matthew D. Hoffman and Andrew Gelman. *The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo*. Nov. 17, 2011.

[14] Stan Development Team. *Stan Reference Manual*.

[15] Aki Vehtari et al. "Rank-normalization, folding, and localization: An improved Rhat for assessing convergence of MCMC". In: *Bayesian Analysis* 16.2 (2021).

[16] Andrew Gelman. "Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)". In: *Bayesian Analysis* 1.3 (2006), pp. 515–534.

[17] Aki Vehtari, Andrew Gelman, and Jonah Gabry. “Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC”. In: *Statistics and Computing* 27.5 (2017), pp. 1413–1432.

Addendum

Stan code

BLMM to analyze current and historical data separately

```
data {
  int<lower=1> N;  // total number of observations
  vector[N] Y;  // response variable
  // data for group-level effects of ID 1
  int<lower=1> N_1;  // number of grouping levels
  int<lower=1> M_1;  // number of coefficients per level
  int<lower=1> J_1[N];  // grouping indicator per observation
  // group-level predictor values
  vector[N] Z_1_1;
}
parameters {
  real Intercept;  // temporary intercept for centered predictors
  real<lower=0> sigma;  // dispersion parameter
  vector<lower=0>[M_1] sd_b;  // group-level standard deviations
  vector[N_1] z_1[M_1];  // standardized group-level effects
}

transformed parameters {
  vector[N_1] r_1_1;  // actual group-level effects
  r_1_1 = (sd_b[1] * (z_1[1]));
}
model {
    vector[N] mu = Intercept + rep_vector(0.0, N);
    for (n in 1:N) {
      // add more terms to the linear predictor
      mu[n] += r_1_1[J_1[n]] * Z_1_1[n];
    }
    target += normal_lpdf(Y | mu, sigma);

    target += normal_lpdf(Intercept| 97.84, 2.5);
    target += student_t_lpdf(sigma | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);
    target += student_t_lpdf(sd_b | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);
    target += std_normal_lpdf(z_1[1]);
}
generated quantities {
  real b_Intercept = Intercept;
```

```
  vector[N] log_lik;
  for (n in 1:N) {
    log_lik[n] = normal_lpdf(Y[n] | Intercept+
    r_1_1[J_1[n]] * Z_1_1[n],sigma);
  }}
```

Application of the Power Prior to univariate data

```
data {
   // Historical data
  int<lower=1> N0;  // total number of observations
  vector[N0] Y0;  // response variable
  // data for group-level effects of ID 1
  int<lower=1> N0_1;  // number of grouping levels
  int<lower=1> M0_1;  // number of coefficients per level
  int<lower=1> J0_1[N0];  // grouping indicator per observation
  // group-level predictor values
  vector[N0] Z0_1_1;

  // Current data
  int<lower=1> N;  // total number of observations
  vector[N] Y;  // response variable
  // data for group-level effects of ID 1
  int<lower=1> N_1;  // number of grouping levels
  int<lower=1> M_1;  // number of coefficients per level
  int<lower=1> J_1[N];  // grouping indicator per observation
  // group-level predictor values
  vector[N] Z_1_1;
}
parameters {
  // Historical data parameters
  real<lower=0> Intercept_0;  // temporary intercept
  vector<lower=0>[M0_1] sd_b_0;  // group-level standard deviations
  vector[N0_1] z0_1[M0_1];  // standardized group-level effects
  real<lower=0,upper=1> a0; //discounting parameter

  real<lower=0> alpha;
  real<lower=0> beta;

  // Current data parameters
  real <lower=0> Intercept;  // temporary intercept
  real<lower=0> sigma;  // dispersion parameter
  vector<lower=0>[M_1] sd_b;  // group-level standard deviations
  vector[N_1] z_1[M_1];  // standardized group-level effects
```

```
}
transformed parameters {
  vector[N_1] r_1_1;  // actual group-level effects
  r_1_1 = (sd_b[1] * (z_1[1]));

   vector[N0_1] r0_1_1;  // actual group-level effects
  r0_1_1 = (sd_b_0[1] * (z0_1[1]));

}
model {
    // initialize linear predictor term
    vector[N0] mu0 = Intercept_0 + rep_vector(0.0, N0);
    for (n0 in 1:N0) {
      // add more terms to the linear predictor
      mu0[n0] += r0_1_1[J0_1[n0]] * Z0_1_1[n0];
    }
    target += a0*normal_lpdf(Y0 | mu0, sigma);
    // initialize linear predictor term
    vector[N] mu = Intercept + rep_vector(0.0, N);
    for (n in 1:N) {
      // add more terms to the linear predictor
      mu[n] += r_1_1[J_1[n]] * Z_1_1[n];
    }
    target += normal_lpdf(Y | mu, sigma);
    target += normal_lpdf(Intercept_0| 97.85, 2.5);
    target += normal_lpdf(Intercept| 97.99, 2.5);
    target += student_t_lpdf(sigma | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);
    target += student_t_lpdf(sd_b_0 | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);
    target += student_t_lpdf(sd_b | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);

    target += beta_lpdf(a0 | alpha,beta);
    target += std_normal_lpdf(z_1[1]);
    target += std_normal_lpdf(z0_1[1]);

    alpha ~ gamma(3,2);
    beta ~ gamma(3,2);

}
generated quantities {
  vector[N] log_lik;
  for (n in 1:N) {
    log_lik[n] = normal_lpdf(Y[n] | Intercept+
```

```
    r_1_1[J_1[n]] * Z_1_1[n], sigma);
  }
}
```

Application of the Commensurate prior to univariate data

```
data {
   // Historical data
  int<lower=1> N0;  // total number of observations
  vector[N0] Y0;  // response variable
  // data for group-level effects of ID 1
  int<lower=1> N0_1;  // number of grouping levels
  int<lower=1> M0_1;  // number of coefficients per level
  int<lower=1> J0_1[N0];  // grouping indicator per observation
  // group-level predictor values
  vector[N0] Z0_1_1;

  // Current data
  int<lower=1> N;  // total number of observations
  vector[N] Y;  // response variable
  // data for group-level effects of ID 1
  int<lower=1> N_1;  // number of grouping levels
  int<lower=1> M_1;  // number of coefficients per level
  int<lower=1> J_1[N];  // grouping indicator per observation
  // group-level predictor values
  vector[N] Z_1_1;
}
parameters {
  // Historical data parameters
  real<lower=0> Intercept_0;  // temporary intercept
  vector<lower=0>[M0_1] sd_b_0;  // group-level standard deviations
  vector[N0_1] z0_1[M0_1];  // standardized group-level effects
  real<lower=0> tau;//commensurability parameter
  real<lower=0> sigma0;  // dispersion parameter

  // Current data parameters
  real <lower=0> Intercept;  // temporary intercept
  real<lower=0> sigma;  // dispersion parameter
  vector<lower=0>[M_1] sd_b;  // group-level standard deviations
  vector[N_1] z_1[M_1];  // standardized group-level effects
}

transformed parameters {
  vector[N_1] r_1_1;  // actual group-level effects
```

```
  r_1_1 = (sd_b[1] * (z_1[1]));


  vector[N0_1] r0_1_1;  // actual group-level effects
  r0_1_1 = (sd_b_0[1] * (z0_1[1]));
  //real<lower=0> tau2;
  //tau2=square(tau);
}

model {
    // initialize linear predictor term
    vector[N0] mu0 = Intercept_0 + rep_vector(0.0, N0);
    for (n0 in 1:N0) {
      // add more terms to the linear predictor
      mu0[n0] += r0_1_1[J0_1[n0]] * Z0_1_1[n0];
    }
    target += normal_lpdf(Y0 | mu0, sigma0);
    // initialize linear predictor term
    vector[N] mu = Intercept + rep_vector(0.0, N);
    for (n in 1:N) {
      // add more terms to the linear predictor
      mu[n] += r_1_1[J_1[n]] * Z_1_1[n];
    }
    target += normal_lpdf(Y | mu, sigma);


    target += normal_lpdf(Intercept_0| 97.85, 2.5);
    target += normal_lpdf(Intercept| 97.99, 2.5);
    target += student_t_lpdf(sigma0 | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);
    target += student_t_lpdf(sd_b_0 | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);
    target += student_t_lpdf(sd_b | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);

    sigma ~ normal(sigma0,tau);
    //tau2 ~ inv_gamma(1,1);
    target += student_t_lpdf(tau | 3, 0, 2.5)
    - student_t_lccdf(0 | 3, 0, 2.5);

    target += std_normal_lpdf(z_1[1]);
    target += std_normal_lpdf(z0_1[1]);

}
```

```
generated quantities {
  vector[N] log_lik;
  for (n in 1:N) {
    log_lik[n] = normal_lpdf(Y[n] | Intercept
    + r_1_1[J_1[n]] * Z_1_1[n],sigma);
  }
}
```

Application of the Mixture prior to univariate data

```
data {
   // Historical data
  int<lower=1> N0;  // total number of observations
  vector[N0] Y0;  // response variable
  // data for group-level effects of ID 1
  int<lower=1> N0_1;  // number of grouping levels
  int<lower=1> M0_1;  // number of coefficients per level
  int<lower=1> J0_1[N0];  // grouping indicator per observation
  // group-level predictor values
  vector[N0] Z0_1_1;
  real<lower=0,upper=1> lambda; //mixing proportion

  // Current data
  int<lower=1> N;  // total number of observations
  vector[N] Y;  // response variable
  // data for group-level effects of ID 1
  int<lower=1> N_1;  // number of grouping levels
  int<lower=1> M_1;  // number of coefficients per level
  int<lower=1> J_1[N];  // grouping indicator per observation
  // group-level predictor values
  vector[N] Z_1_1;
}
parameters {
  // Historical data parameters
  real<lower=0> Intercept_0;  // temporary intercept
  vector<lower=0>[M0_1] sd_b_0;  // group-level standard deviations
  vector[N0_1] z0_1[M0_1];  // standardized group-level effects
  real<lower=0> sigma0;  // dispersion parameter
  // Current data parameters
  real <lower=0> Intercept;  // temporary intercept
  real<lower=0> sigma;  // dispersion parameter
  vector<lower=0>[M_1] sd_b;  // group-level standard deviations
  vector[N_1] z_1[M_1];  // standardized group-level effects
}
```

```
transformed parameters {
  vector[N_1] r_1_1;  // actual group-level effects
  r_1_1 = (sd_b[1] * (z_1[1]));
  vector[N0_1] r0_1_1;  // actual group-level effects
  r0_1_1 = (sd_b_0[1] * (z0_1[1]));
}
model {
  // initialize linear predictor term
  vector[N0] mu0 = Intercept_0 + rep_vector(0.0, N0);
  for (n0 in 1:N0) {
  // add more terms to the linear predictor
  mu0[n0] += r0_1_1[J0_1[n0]] * Z0_1_1[n0];
  }
  target += normal_lpdf(Y0 | mu0, sigma0);

  // initialize linear predictor term
  vector[N] mu = Intercept + rep_vector(0.0, N);
  for (n in 1:N) {
  // add more terms to the linear predictor
  mu[n] += r_1_1[J_1[n]] * Z_1_1[n];
  }
  target += normal_lpdf(Y | mu, sigma);

  target += normal_lpdf(Intercept_0| 97.85, 2.5);
  target += normal_lpdf(Intercept| 97.99, 2.5);
  target += student_t_lpdf(sigma0 | 3, 0, 2.5)
  - student_t_lccdf(0 | 3, 0, 2.5);
  target += student_t_lpdf(sd_b_0 | 3, 0, 2.5)
  - student_t_lccdf(0 | 3, 0, 2.5);
  target += student_t_lpdf(sd_b | 3, 0, 2.5)
  - student_t_lccdf(0 | 3, 0, 2.5);
  //defintion of the mixture prior
  target += log_sum_exp(log(lambda)
  + normal_lpdf(sigma | sigma0, 0.3),
  log(1-lambda) + normal_lpdf(sigma | 0, 2.5));

  target += std_normal_lpdf(z_1[1]);
  target += std_normal_lpdf(z0_1[1]);

}
generated quantities {
  vector[N] log_lik;
  for (n in 1:N) {
    log_lik[n] = normal_lpdf(Y[n] | Intercept
    + r_1_1[J_1[n]] * Z_1_1[n],sigma);}}
```

BLMM for bivariate potency data

```
data{
  int<lower=1> n; //nr. of samples
  int<lower=1> i; //nr. of batches
  int<lower=1> k; //nr. of assays
  int<lower=1,upper=i> j[n]; //batch ID for each outcome
  vector[n] y1; //outcome 1
  vector[n] y2; //outcome 2
  cov_matrix[k] S;//covariance matrix used in inv-Wishart prior
  cov_matrix[k] R;//covariance matrix used in inv-Wishart prior
}
transformed data {
  vector[k] y[n];  // response array
  for (x in 1:n) {
    y[x] = transpose([y1[x],y2[x]]);
  }
}
parameters{
  vector[k] a; //population process means for API1 and API2
  vector[k] b[i];//random batch intercepts for API1 and API2

  cov_matrix[k] Sigma;
  cov_matrix[k] Sigma_R;
}
model{
  //level-2 likelihood
  b ~ multi_normal(a,Sigma_R);
  vector[k] mu[n];

  for (x in 1:n) {
    mu[x] = b[j[x]];
    }

        //level-1 likelihood
        y ~ multi_normal(mu,Sigma);

  //Priors
  a[1] ~ normal(99.45,2.5);
        a[2] ~ normal(97.85,2.5);

  Sigma ~ inv_wishart(4,S);
  Sigma_R ~ inv_wishart(4,R);

}
```

```
generated quantities {
  real sigma11= Sigma[1,1];
  real sigma22=Sigma[2,2];
  real sigma12=Sigma[1,2];
  vector[n] log_lik;
  for (x in 1:n) {
    log_lik[x] = multi_normal_lpdf(y[x] | b[j[x]],Sigma);
  }
}
```

Application of the Power prior on bivariate potency data

```
data{
  int<lower=1> n_0; //nr. of samples
  int<lower=1> i_0; //nr. of batches
  int<lower=1> k; //nr. of assays
  int<lower=1,upper=i_0> j_0[n_0]; //batch ID for each outcome
  vector[n_0] y1_0; //outcome 1
  vector[n_0] y2_0; //outcome 2
  cov_matrix[k] S;//covariance matrix used in inv-Wishart prior
  cov_matrix[k] R_0;//covariance matrix used in inv-Wishart prior

  int<lower=1> n; //nr. of samples
  int<lower=1> i; //nr. of batches
  int<lower=1,upper=i> j[n]; //batch ID for each outcome
  vector[n] y1; //outcome 1
  vector[n] y2; //outcome 2
  cov_matrix[k] R;//covariance matrix used in inv-Wishart prior
}
transformed data {
  vector[k] y[n];  // response array current data
  for (x in 1:n) {
    y[x] = transpose([y1[x],y2[x]]);
  }
  vector[k] y_0[n_0]; // response array historical data
  for (x in 1:n_0) {
    y_0[x] = transpose([y1_0[x],y2_0[x]]);
  }
}
parameters{
  vector[k] a_0; //population process means for hist. data
  vector[k] b_0[i_0];//random batch intercepts for API1 and API2

  vector[k] a; //population process means for current data
```

```
  vector[k] b[i];//random batch intercepts for API1 and API2

  cov_matrix[k] Sigma;
  cov_matrix[k] Sigma_R_0;
  cov_matrix[k] Sigma_R;

  real<lower=0,upper=1> a0; //discounting parameter

  real<lower=0> alpha;
  real<lower=0> beta;
}
model{
  //level-2 likelihood for historical data
  b_0 ~ multi_normal(a_0,Sigma_R_0);
  vector[k] mu_0[n_0];

   for (x in 1:n_0) {
      mu_0[x] = b_0[j_0[x]];
    }
        target += a0*multi_normal_lpdf(y_0 | mu_0, Sigma);

        //level-1 likelihood for current data
  b ~ multi_normal(a,Sigma_R);

  vector[k] mu[n];

  for (x in 1:n) {
      mu[x] = b[j[x]];
    }
        target += multi_normal_lpdf(y | mu, Sigma);

  //Priors
  a_0[1] ~ normal(99.45,2.5);
        a_0[2] ~ normal(97.85,2.5);

        a[1] ~ normal(100.42,2.5);
        a[2] ~ normal(97.99,2.5);

  Sigma ~ inv_wishart(4,S);
  Sigma_R_0 ~ inv_wishart(4,R_0);
  Sigma_R ~ inv_wishart(4,R);

  target += beta_lpdf(a0 | alpha,beta);
  alpha ~ gamma(3,2);
  beta ~ gamma(3,2);
```

```
}
generated quantities {
  real sigma11= Sigma[1,1];
  real sigma22=Sigma[2,2];
  real sigma12=Sigma[1,2];
  vector[n] log_lik;
  for (x in 1:n) {
    log_lik[x] = multi_normal_lpdf(y[x] | b[j[x]],Sigma);
  }
}
```

Application of the Commensurate prior to bivariate potency data

```
data{
  int<lower=1> n_0; //nr. of samples
  int<lower=1> i_0; //nr. of batches
  int<lower=1> k; //nr. of assays
  int<lower=1,upper=i_0> j_0[n_0]; //batch ID for each outcome
  vector[n_0] y1_0; //outcome 1
  vector[n_0] y2_0; //outcome 2
  cov_matrix[k] S_0;//covariance matrix used in inv-Wishart prior
  cov_matrix[k] R_0;//covariance matrix used in inv-Wishart prior

  int<lower=1> n; //nr. of samples
  int<lower=1> i; //nr. of batches
  int<lower=1,upper=i> j[n]; //batch ID for each outcome
  vector[n] y1; //outcome 1
  vector[n] y2; //outcome 2
  cov_matrix[k] R;//covariance matrix used in inv-Wishart prior
}
transformed data {
  vector[k] y[n];  // response array current data
  for (x in 1:n) {
    y[x] = transpose([y1[x],y2[x]]);
  }
  vector[k] y_0[n_0]; // response array historical data
  for (x in 1:n_0) {
    y_0[x] = transpose([y1_0[x],y2_0[x]]);
  }
}
parameters{
  vector[k] a_0; //population process means for API1 and API2
  vector[k] b_0[i_0];//random batch intercepts for API1 and API2
```

```
  vector[k] a; //population process means for API1 and API2
  vector[k] b[i];//random batch intercepts for API1 and API2

  cov_matrix[k] Sigma_0;
  cov_matrix[k] Sigma_R_0;
  cov_matrix[k] Sigma_R;
  cov_matrix[k] Sigma;
  real<lower=4> tau;
}
transformed parameters {
  real lambda;
  lambda=1/tau;
}
model{
  //level-2 likelihood for historical data
  b_0 ~ multi_normal(a_0,Sigma_R_0);
  vector[k] mu_0[n_0];

  for (x in 1:n_0) {
      mu_0[x] = b_0[j_0[x]];
    }
        target += multi_normal_lpdf(y_0 | mu_0, Sigma_0);

        //level-2 likelihood for current data
  b ~ multi_normal(a,Sigma_R);
  vector[k] mu[n];

  for (x in 1:n) {
      mu[x] = b[j[x]];
    }
        //level-1 likelihood for current data
        target += multi_normal_lpdf(y | mu, Sigma);

  //Priors
  a_0[1] ~ normal(99.45,2.5);
        a_0[2] ~ normal(97.85,2.5);

        a[1] ~ normal(100.42,2.5);
        a[2] ~ normal(98,2.5);

  Sigma_0 ~ inv_wishart(4,S_0);
  Sigma_R_0 ~ inv_wishart(4,R_0);
  Sigma_R ~ inv_wishart(4,R);
  Sigma ~ inv_wishart(tau,Sigma_0);
```

```
  lambda ~ beta(1,1);
}
generated quantities {
  real sigma11= Sigma[1,1];
  real sigma22=Sigma[2,2];
  real sigma12=Sigma[1,2];
  vector[n] log_lik;
  for (x in 1:n) {
    log_lik[x] = multi_normal_lpdf(y[x] | b[j[x]],Sigma);
  }
}
```

R code

Univariate potency dependent sample size calculation

```
OCurve4Assay <- function(StanModel, ModelName, SampleSize, Specs,
Confidence, Coverage, SeedNum){

  #===>Tolerance interval parameters
  Betat  <- Coverage
  Gammat <- Confidence
  #===>Specifications
  specs <- Specs

  PosteriorSamples <- rstan::extract(StanModel)
  K <- SampleSize
  Kmax <- max(K)

  ### posterior distribution of residual SD
  RandomError  <- PosteriorSamples$sigma
  X1   <- matrix(NA, ncol = Kmax, nrow = length(RandomError))

  set.seed(SeedNum)
  for(i in 1:length(RandomError)){
    X1[i,] <- rnorm(Kmax, 0, RandomError[i])
  }


  if (file.exists(paste("Results/",
    paste0(ModelName,"_ETM.rds",sep=""), sep=""))) {
    ### recall the tolerance object:
    ETM <- readRDS(paste("Results/",
```

```
  paste0(ModelName,"_ETM.rds",sep=""), sep="")) } else {
    ### Residual Tolerance intervals:
    set.seed(SeedNum)

    ETM <- NULL
    for (j in 1:length(K)){
      etm <- matrix(NA, nrow= length(RandomError), ncol=3)
      etm[,1] <- K[j]
      kvalue <- K.factor(K[j], alpha = (1 - Gammat),
      P = Betat, side = 2, method = "EXACT", m = 50)
      for (i in 1:length(RandomError)){
        etm[i, 2:3] <- as.matrix(mean(X1[i,1:K[j]])
        +c(-1,1)*sd(X1[i,1:K[j]])*kvalue)
        if(i %in% seq(0, length(RandomError), by=200)){
          cat("N =", K[j], "(",j,"/", length(K),"),", "chain(",i, "/",
              length(RandomError), ")", "\n")
        }
      }
      ETM <- rbind(ETM, etm)
    }
    ### save the tolerance object:
    saveRDS(ETM, paste("Results/",
    paste0(ModelName,"_ETM.rds",sep=""), sep=""))


  }


Betas <- seq(from = specs[1], to = specs[2], by=0.1)
POS <- NULL
for (beta in Betas){
  ETMData <- as.data.frame(ETM)
  ETMData$mean <- beta
  names(ETMData) <- c("k","lower","upper","mean")
  ETMData$lower <- ETMData$lower+beta
  ETMData$upper <- ETMData$upper+beta
  ETMData$success <- ifelse(((ETMData$lower > specs[1]
  )&(ETMData$upper < specs[2])),1,0)

  pos <- ETMData %>%
    group_by(mean,k) %>%
    dplyr::summarize(pos=mean(success))

  POS <- rbind(POS,pos)
```

```
}

### save object for success/failure for the true batch means
(tolerance intervals):
saveRDS(POS, paste("Results/",
paste0(ModelName,"_POS.rds",sep=""), sep=""))

#===> recall the tolerance object:
ETM <- readRDS(paste("Results/",
paste0(ModelName,"_ETM.rds",sep=""), sep=""))

### read success/failure of true batch means (tolerance intervals):
POS <- readRDS(paste("Results/",
paste0(ModelName,"_POS.rds",sep=""), sep=""))

#===> Results graphs
#===> generate the distribution of a future random batch:
Rbetas <- NULL
RanFun <- function(x,y){
  rbetas <- rnorm(1, mean = x, sd = y)
  Rbetas <- rbind(Rbetas, rbetas)
  return(Rbetas)
}

#===> Process mean
ProcessMean  <- PosteriorSamples$Intercept
Mu_ProcessMean <- mean(ProcessMean)
SigmaBatch <- PosteriorSamples$sd_b
RandomBatchMean <- mapply(RanFun, ProcessMean , SigmaBatch)
Process_Mu_CL <- quantile(RandomBatchMean,c(0.025,0.975))

OCurvePlot <- POS %>% mutate(n=as.factor(k))  %>%
  ggplot(.,aes(x = mean,y = pos, group = n, col = n))+
  geom_line() +
  labs(x = "True batch mean",
       y = "PoS",
       title = ModelName) +
  scale_y_continuous(breaks = seq(0, 1, 0.05)) +
  scale_x_continuous(breaks = seq(specs[1], specs[2], 1)) +
  geom_vline(aes(xintercept = Mu_ProcessMean),
  linetype  = "dashed", col = "blue") +
  geom_hline(aes(yintercept = 0.95),  col = "red") +
  annotate("text", x = (Mu_ProcessMean+0.2),
  y = 0.45, label = "Process mean",
  color = "blue", angle = 90) +
```

```
    #Shaded area
    annotate("rect",ymin = -Inf, ymax = Inf, xmin = Process_Mu_CL[1],
    xmax = Process_Mu_CL[2], alpha = .1, fill = "brown")+
    annotate("text", x = (Process_Mu_CL[1]-0.2),
    y = 0.45, label = "Lower 95% PL",
    color = "brown", angle = 90) +
    annotate("text", x = (Process_Mu_CL[2]+0.2),
    y = 0.45, label = "Upper 95% PL",
             color = "brown", angle = 90) +
    theme_bw() +
    theme(axis.ticks = element_blank(),
          # legend.position = "bottom",
          axis.text.x = element_text(angle = 90,
                                     hjust = 1,
                                     vjust = 0.5),
          plot.title = element_text(hjust = 0.5))
    return(OCurvePlot)
}
```

Bivariate potency based sample size calculation

```
Samplesize.multivariate <- function(StanModel,
ModelName, SampleSize=seq(3:20), Specs, Confidence, Coverage, SeedNum){


  #===>Tolerance interval parameters
  Betat  <- Coverage
  Gammat <- Confidence
  #===>Specifications
  specs <- Specs


  PosteriorSamples <-as_draws_df(StanModel$draws())
  K <- SampleSize
  Kmax <- max(K)


  ### posterior distrn of residual SD
  RandomError1  <- unlist(PosteriorSamples[,"sigma11"])
  RandomError2  <- unlist(PosteriorSamples[,"sigma22"])
  RandomError12 <- unlist(PosteriorSamples[,"sigma12"])

  X1   <- matrix(NA, ncol = Kmax, nrow = length(RandomError1))
  X2   <- matrix(NA, ncol = Kmax, nrow = length(RandomError2))
```

```
set.seed(SeedNum)
for(i in 1:length(RandomError1)){
  container <- mvrnorm(n=Kmax,mu=c(0,0),
  Sigma=matrix(c(RandomError1[i],RandomError12[i],
  RandomError12[i],RandomError2[i]),
  nrow = 2, ncol = 2, byrow = TRUE))
  X1[i,] <- container[,1]
  X2[i,] <- container[,2]
}

ETM1 <- NULL
for (j in 1:length(K)){
  etm1 <- matrix(NA, nrow= length(RandomError1), ncol=3)
  etm1[,1] <- K[j]
  kvalue <- K.factor(K[j], alpha = (1 - Gammat),
  P = Betat, side = 2, method = "EXACT", m = 50)
  for (i in 1:length(RandomError1)){
    etm1[i, 2:3] <- as.matrix(mean(X1[i,1:K[j]]) +
    c(-1,1)*sd(X1[i,1:K[j]])*kvalue)
    if(i %in% seq(0, length(RandomError1), by=200)){
      cat("N =", K[j], "(",j,"/", length(K),"),",
      "chain(",i, "/",
          length(RandomError1), ")", "\n")
    }
  }
  ETM1 <- rbind(ETM1, etm1)
}

ETM2 <- NULL
for (j in 1:length(K)){
  etm2 <- matrix(NA, nrow= length(RandomError2), ncol=3)
  etm2[,1] <- K[j]
  kvalue <- K.factor(K[j], alpha = (1 - Gammat),
  P = Betat, side = 2, method = "EXACT", m = 50)
  for (i in 1:length(RandomError2)){
    etm2[i, 2:3] <- as.matrix(mean(X2[i,1:K[j]]) +
    c(-1,1)*sd(X2[i,1:K[j]])*kvalue)
    if(i %in% seq(0, length(RandomError2), by=200)){
      cat("N =", K[j], "(",j,"/", length(K),"),", "chain(",i, "/",
          length(RandomError2), ")", "\n")
    }
  }
  ETM2 <- rbind(ETM2, etm2)
}
```

```
  Beta1 <- seq(from = specs[1], to = specs[2], by=0.1)
  Beta2 <- seq(from = specs[1], to = specs[2], by=0.1)

  POS1 <- NULL
  for (i in Beta1){
    ETMData <- as.data.frame(ETM1)
    ETMData$mean1 <- i
    names(ETMData) <- c("k","lower1","upper1","mean1")
    ETMData$lower1 <- ETMData$lower1+i
    ETMData$upper1 <- ETMData$upper1+i
    ETMData$success <- ifelse(((ETMData$lower1 > specs[1]
    )&(ETMData$upper1 < specs[2])),1,0)

    pos1 <- ETMData %>%
      group_by(mean1,k) %>%
      dplyr::summarize(pos1=mean(success))

    POS1 <- rbind(POS1,pos1)}

  POS2 <- NULL
  for (i in Beta2){
    ETMData <- as.data.frame(ETM2)
    ETMData$mean2 <- i
    names(ETMData) <- c("k","lower2","upper2","mean2")
    ETMData$lower2 <- ETMData$lower2+i
    ETMData$upper2 <- ETMData$upper2+i
    ETMData$success <- ifelse(((ETMData$lower2 > specs[1]
    )&(ETMData$upper2 < specs[2])),1,0)

    pos2 <- ETMData %>%
      group_by(mean2,k) %>%
      dplyr::summarize(pos2=mean(success))

    POS2 <- rbind(POS2,pos2)}


  POS <- merge(POS1,POS2, by.x = "k",
               by.y = "k", all.x = TRUE, all.y = TRUE)
  POS$pos <- POS$pos1*POS$pos2

  results <- POS %>% filter(mean1==100.4,mean2==98)

  return(results)
}
```

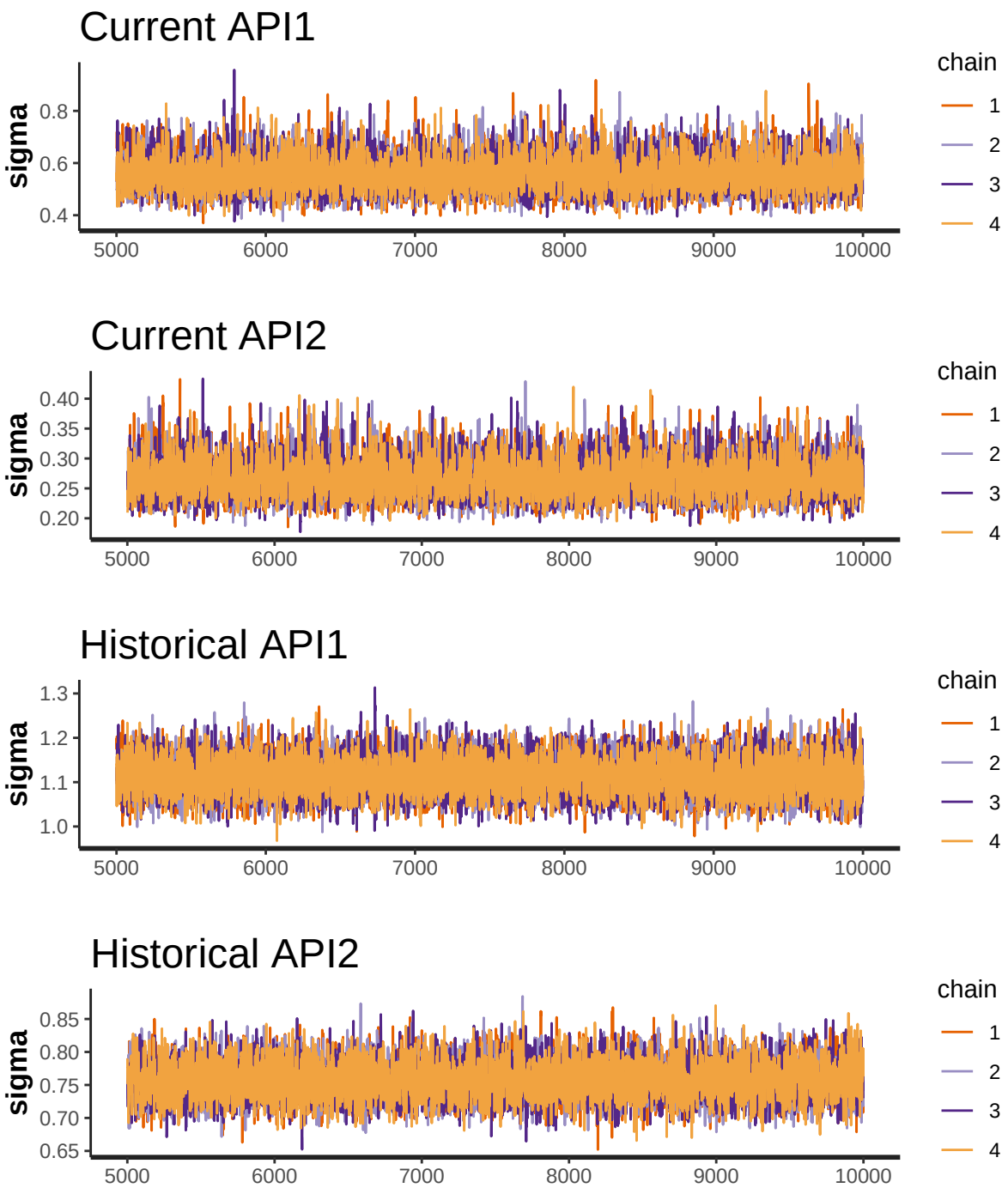

Figure 5: Traceplots for σ_0 and σ_1 when the univariate data are modeled separately

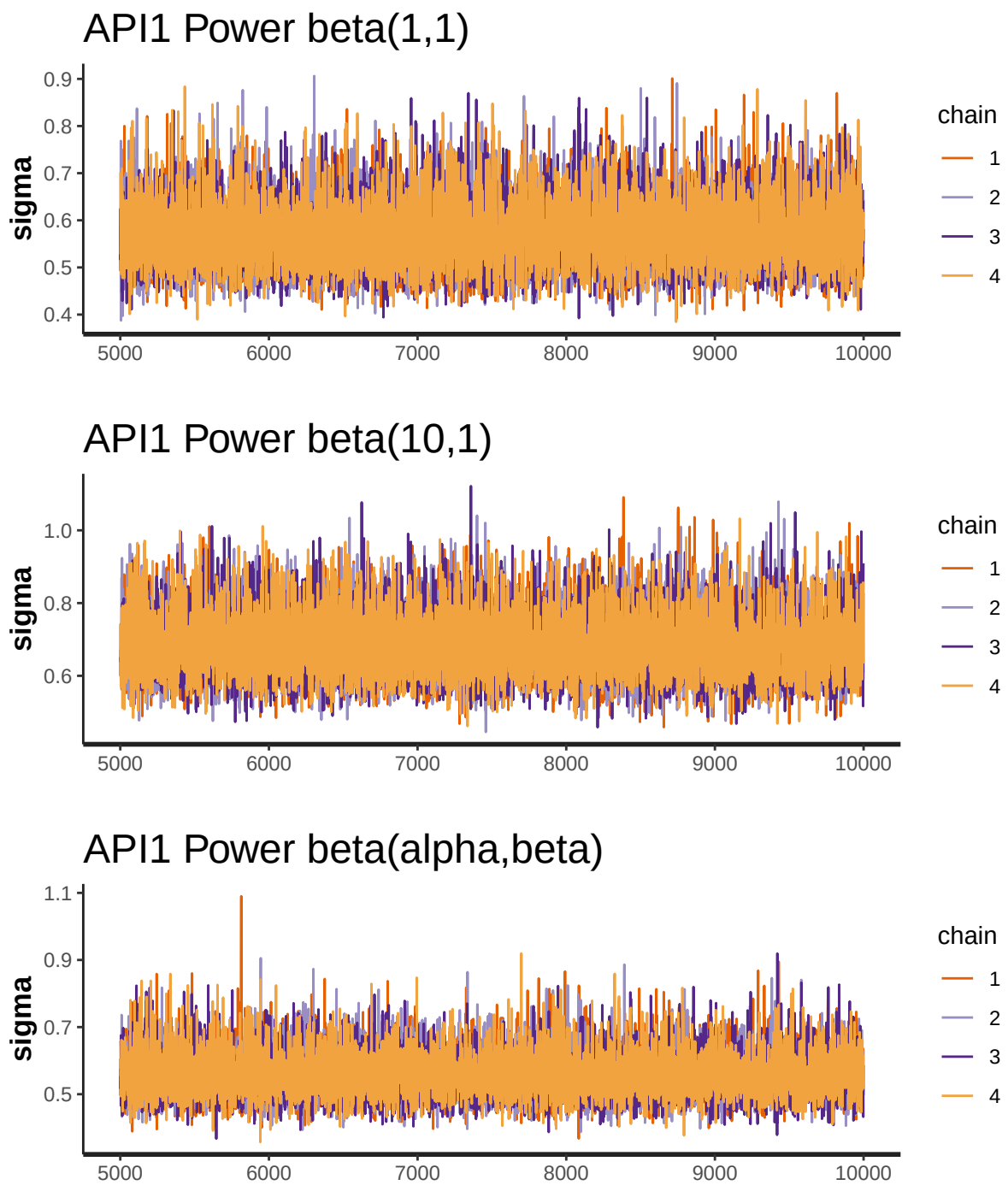

Figure 6: Traceplots for σ after applying the Power prior

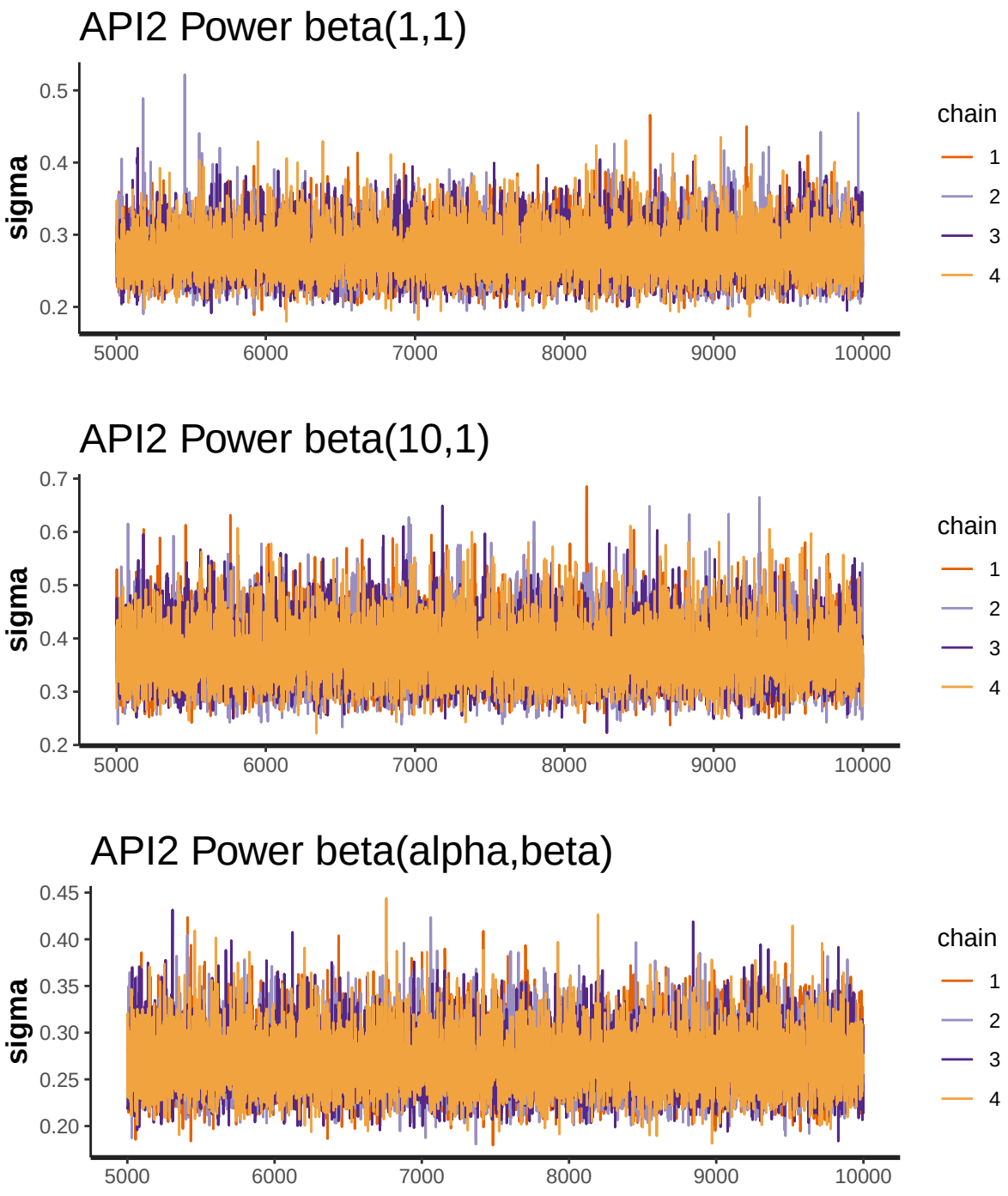

Figure 7: Traceplots for σ after applying the Power prior

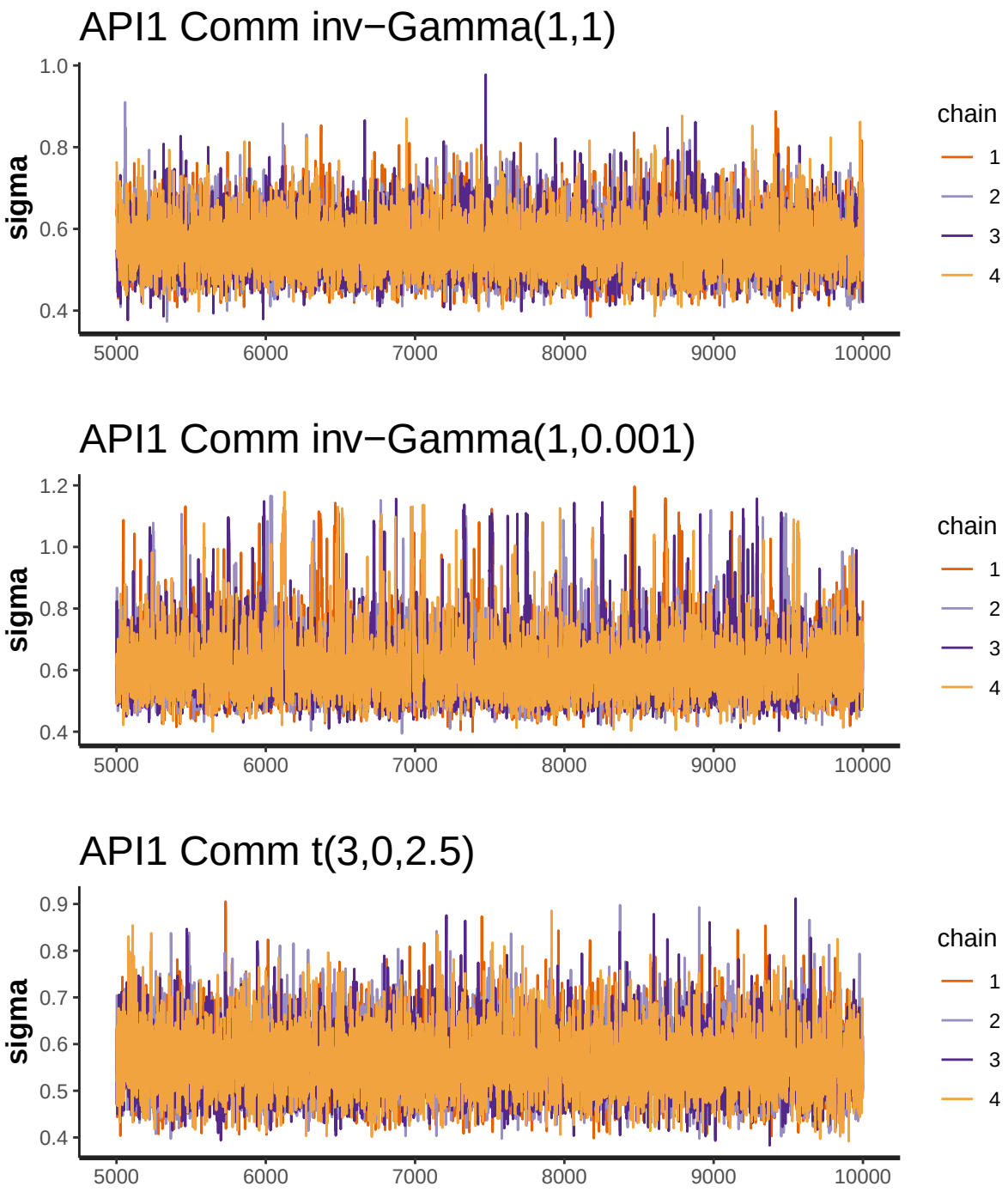

Figure 8: Traceplots for σ_1 after applying the Commensurate prior

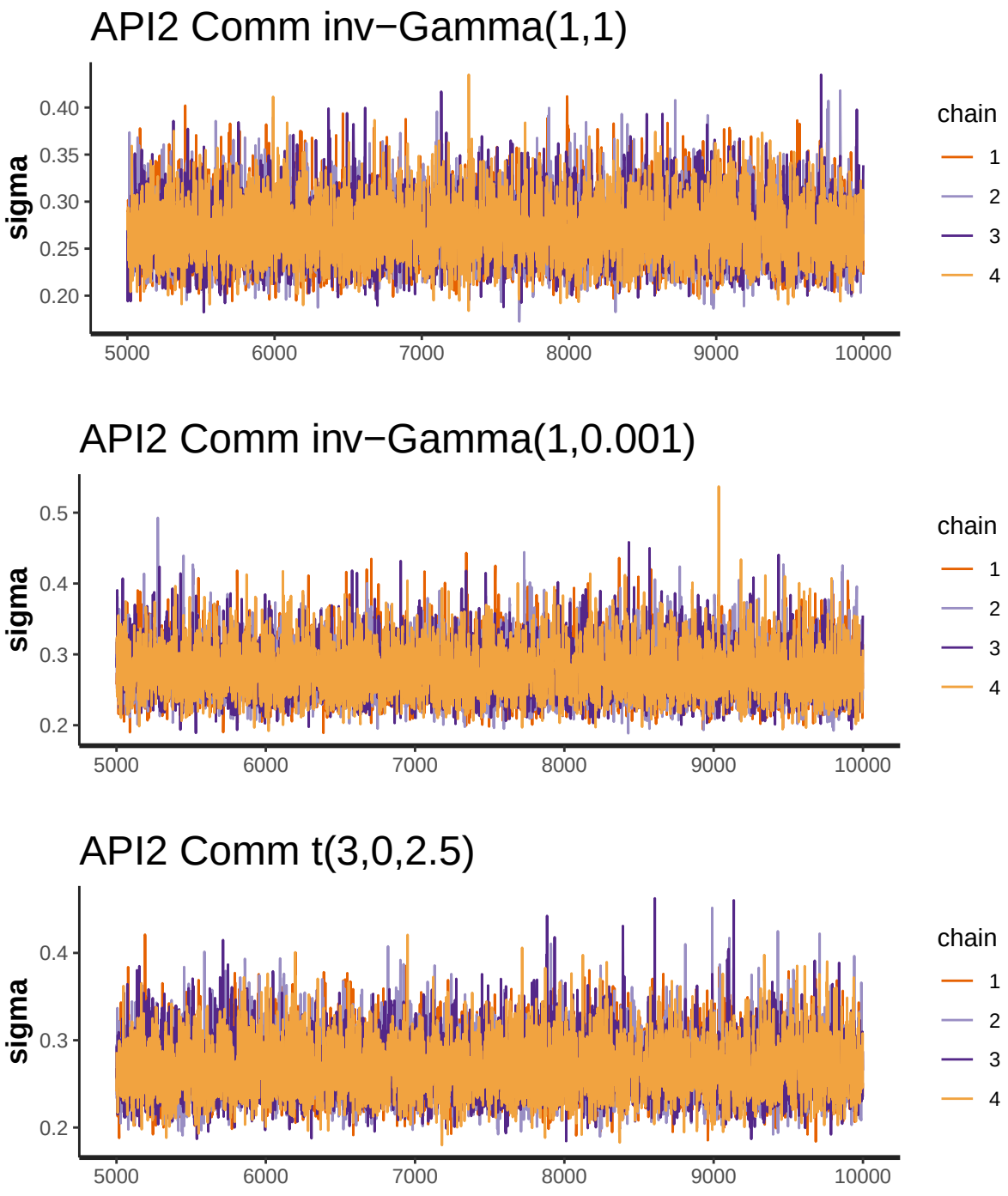

Figure 9: Traceplots for σ_1 after applying the Commensurate prior

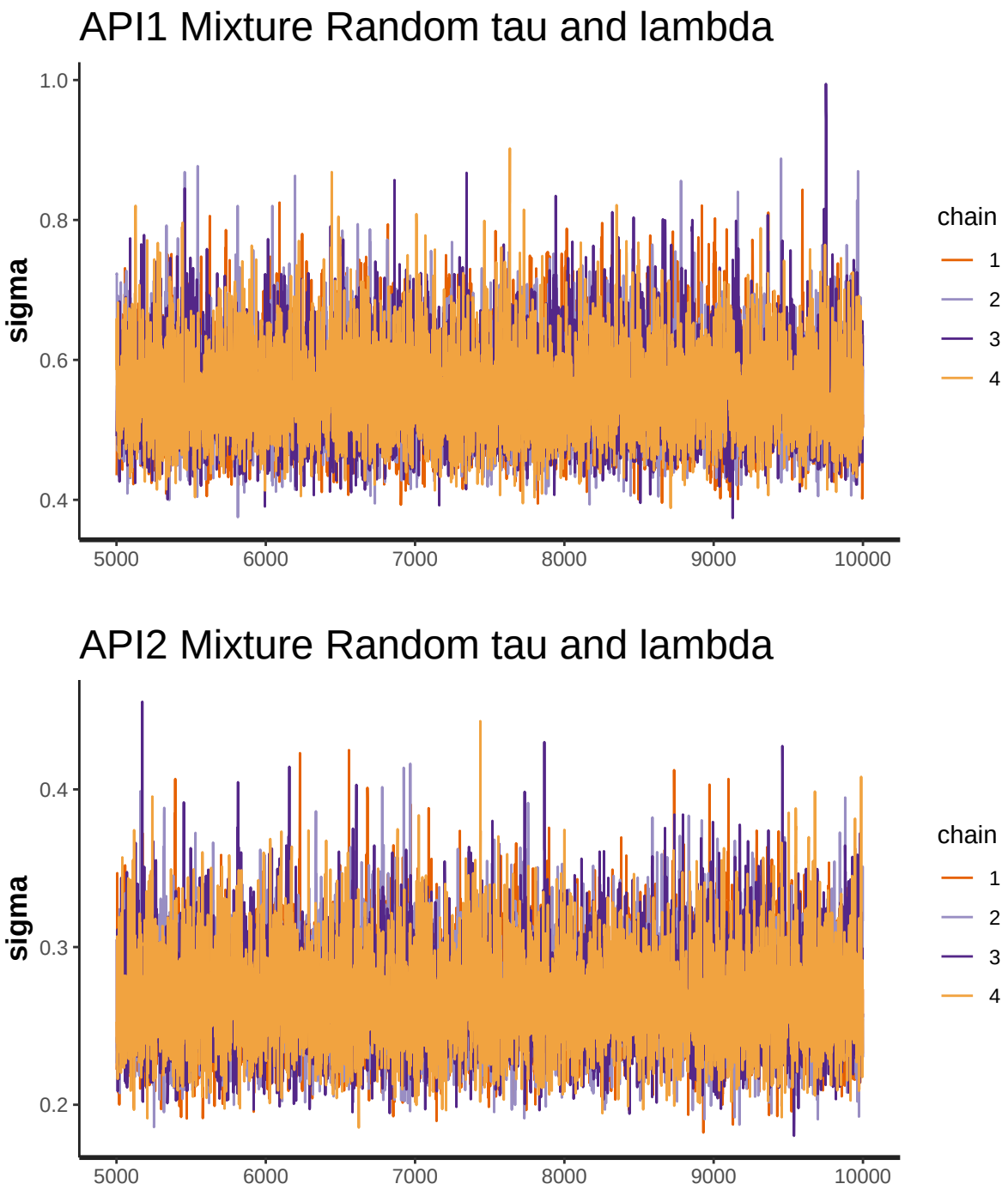

Figure 10: Traceplots for σ_1 after applying the Mixture prior with random τ and λ

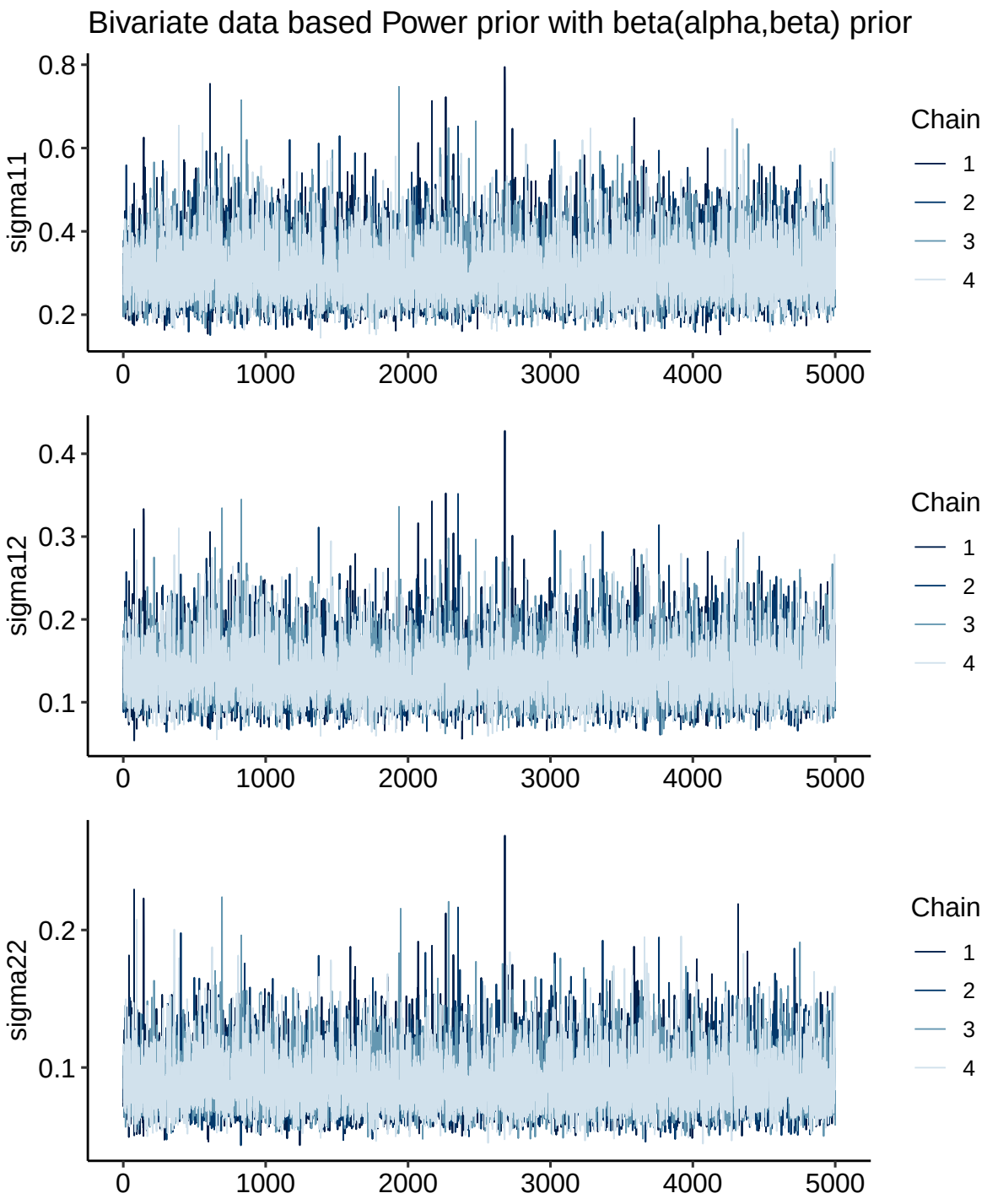

Figure 11: Traceplots for the variance and covariance after applying the Power prior

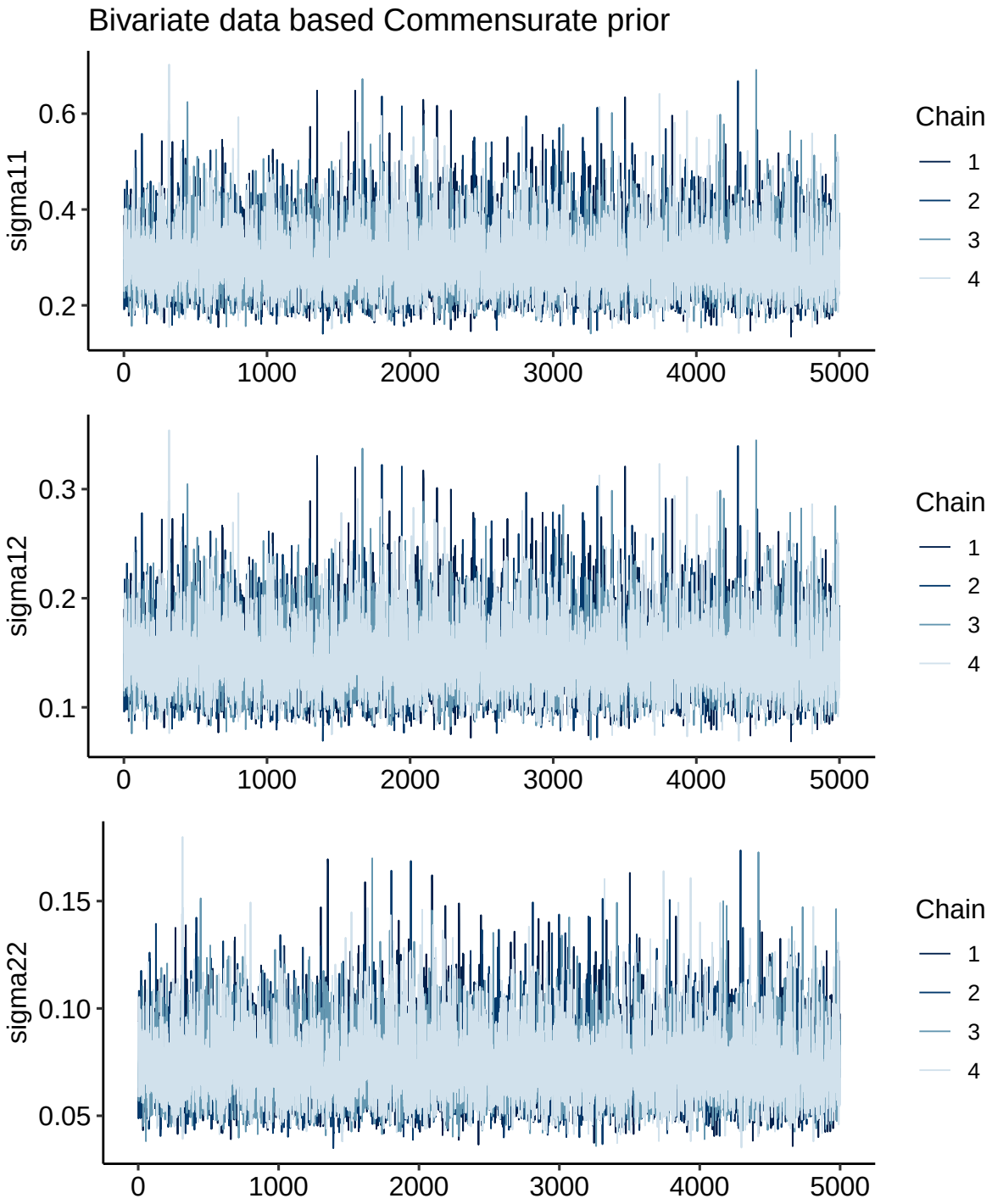

Figure 12: Traceplots for the variance and covariance after applying the Commensurate prior