Arlenda

Advance your business with statistics

Manufacturing process simulation: from bioreactor to shelf-life using Bayesian predictions

BAYES 2016 LEUVEN

17-20 MAY 2016

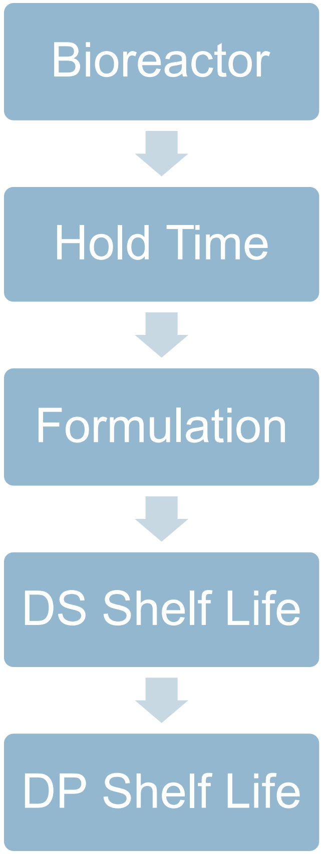
Pierre Lebrun and Perceval Sondag

Contact:
pierre.lebrun@arlenda.com

Outline

- The Manufacturing Process and Multiple Uncertainties
- A Bayesian approach
- Prediction of the Complete Process
  - Modeling each step from historical data
  - Posterior predictive distribution
- Sensitivity to Process and Formulation Parameters
- Take Home Messages

Process Steps



- Bioreactor
  - Process parameter variability
  - Model parameter estimate uncertainty
- In Process Hold Times
  - Slope estimate uncertainty
  - Duration variability
- Formulation
  - Formulation parameter variability
- Shelf life
  - Degradation rate estimate uncertainty

+

Measurement error

Batch/Run/… effect

Model uncertainty

…

GUM methodologies

- The idea is to combine systematic and random errors of a process

  Here, what we do is a strong parallelism to GUM

  In GUM, uncertainties are standardized (e.g. % loss) and combined for instance using Gaussian properties

$$\begin{array}{l} Y \sim N(\mu_Y, \sigma_Y^2) \\ X \sim N(\mu_X, \sigma_X^2) \end{array} \longrightarrow Z = X + Y; \longrightarrow Z \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2).$$

- What lacks in GUM is the uncertainty of the… uncertainty measurements, easily handled in the Bayesian framework

Bayesian manifesto

- Why a Bayesian approach ?

  Because we want to predict (outcome of the process steps)

  Because we want to make probabilistic statements of an outcome

  -> P(success) or P(OOS)

  Because we may (sometimes) have prior knowledge

  Because, thanks to MCMC simulations, we can handle simple to very complex models in a unified framework (yes, speed of implementation matters more than running speed of the samplers)

  In general, models are pretty simple. e.g. two-way random ANOVA models… but with unbalanced data, prediction as a frequentist is already not a good option…

  Because, thanks to Monte-Carlo methods, I can pool and propagate all uncertainties from the beginning to the end of the process

  Why focus on maximum likelihood when we can play with all the posterior distribution ?

  Because we want to predict

Objective: Simulate the Complete Process How ?

1. Modeling

For each step, fit a Bayesian Generalized Multivariate Linear Mixed Model based on historical data

Example:

$$\mathrm{g}(\boldsymbol{y}) = \boldsymbol{X\beta} + \boldsymbol{Z\gamma} + \boldsymbol{\epsilon}$$

With $\boldsymbol{\gamma}$ the random effects (e.g. run, batch), and $\boldsymbol{\epsilon}$ the measurement error.

Uncertainties in $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ due to measurement and modelling

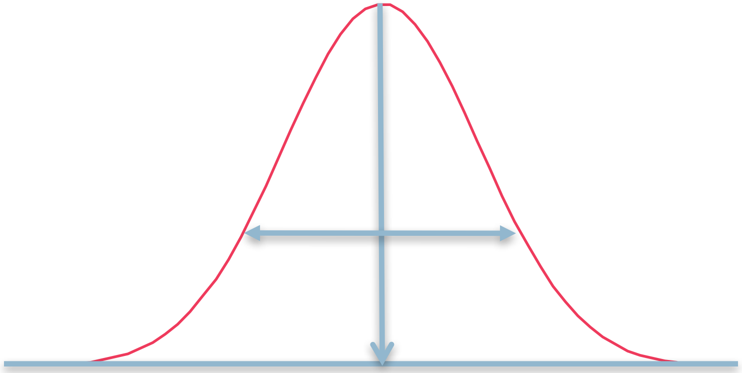
→ **Impact on prediction !**

→ **To obtain "good" prediction, $\boldsymbol{\epsilon}$ must be small, and $\boldsymbol{\beta}$** and $\boldsymbol{\gamma}$ must be estimated with quality (small posterior uncertainty)

How to make predictions

Monte-Carlo Simulations

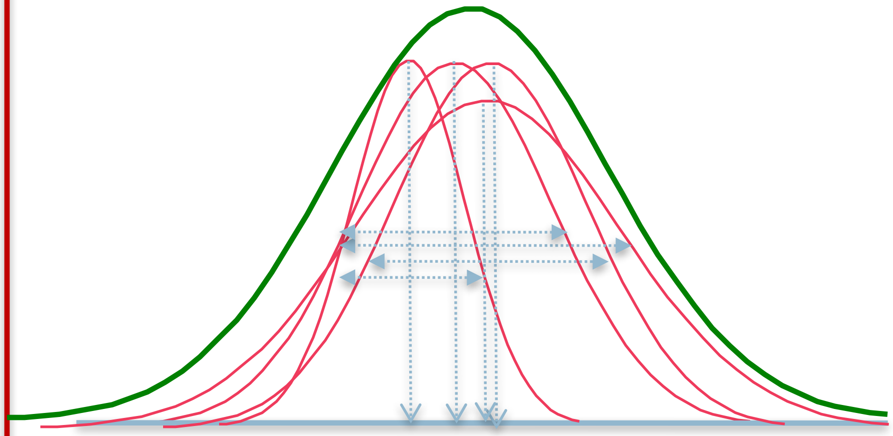
where the “new observations” are drawn from distribution “centered” on estimated location and dispersion parameters (treated wrongly as “true values”).



Predictions

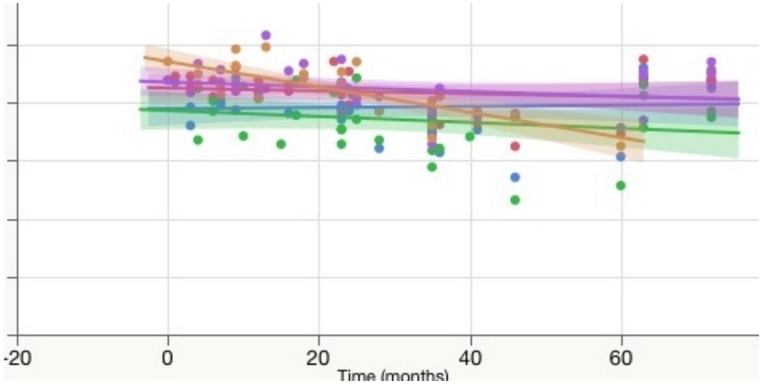
First, by drawing a mean and a variance from the posteriors and, second, drawing an observation from resulting distribution

$$p(\tilde{y} \mid \mathbf{y}) = \int_{\boldsymbol{\theta}} p(\tilde{y} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{y})\, d\boldsymbol{\theta}$$



MCMCglmm, small code example

```
### Model
require(MCMCglmm)
m = MCMCglmm(fixed = cbind(y1,y2) ~ trait + trait:time +
                                trait:factor1 + trait:factor1:time- 1,
                         data = data,
                         family = c("gaussian", “gaussian" ),
                         rcov = ~ us(trait):units,
                         random = ~ idh(trait):Run + idh(trait):Site,
                         prior=list(R=list(V=R_scale,nu=3),
                                    G=list(G1=list(V=diag(Run_scale),nu=3),
                                    G2=list(V=diag(Site_scale),nu=3))),
                         nitt = 130000,
                         burnin = 30000,
                         thin=10)
```



MCMCglmm, small code example

```
### Model
require(MCMCglmm)
m = MCMCglmm(fixed = cbind(y1,y2) ~ trait + trait:time +
                                         trait:factor1 + trait:factor1:time- 1,
                              data = data,
                              family = c("gaussian", “gaussian" ),
                              rcov = ~ us(trait):units,
                              random = ~ idh(trait):Run + idh(trait):Site,
                              prior=list(R=list(V=R_scale,nu=3),
                                         G=list(G1=list(V=diag(Run_scale),nu=3),
                                         G2=list(V=diag(Site_scale),nu=3))),
                              nitt = 130000,
                              burnin = 30000,
                              thin=10)
```

Singular syntax for fixed effect formula

MCMCglmm, small code example

```
### Model
require(MCMCglmm)
m = MCMCglmm(fixed = cbind(y1,y2) ~ trait + trait:time +
                                          trait:factor1 + trait:factor1:time- 1,
                             data = data,
                             family = c("gaussian", "gaussian"),
                             rcov = ~ us(trait):units,
                             random = ~ idh(trait):Run + idh(trait):Site,
                             prior=list(R=list(V=R_scale,nu=3),
                                        G=list(G1=list(V=diag(Run_scale),nu=3),
                                        G2=list(V=diag(Site_scale),nu=3))),
                             nitt = 130000,
                             burnin = 30000,
                             thin=10)
```

Can handle several link functions

MCMCglmm, small code example

```
### Model
require(MCMCglmm)
m = MCMCglmm(fixed = cbind(y1,y2) ~ trait + trait:time +
                                  trait:factor1 + trait:factor1:time- 1,
                        data = data,
                        family = c("gaussian", “gaussian" ),
                        rcov = ~ us(trait):units,
                        random = ~ idh(trait):Run + idh(trait):Site,
                        prior=list(R=list(V=R_scale,nu=3),
                                 G=list(G1=list(V=diag(Run_scale),nu=3),
                                 G2=list(V=diag(Site_scale),nu=3))),
                        nitt = 130000,
                        burnin = 30000,
                        thin=10)
```

Estimate the covariance of residuals and handle multiple random effects, **even unbalanced**

MCMCglmm, small code example

```
### Model
require(MCMCglmm)
m = MCMCglmm(fixed = cbind(y1,y2) ~ trait + trait:time +
                                 trait:factor1 + trait:factor1:time- 1,
             data = data,
             family = c("gaussian", “gaussian" ),
             rcov = ~ us(trait):units,
             random = ~ idh(trait):Run + idh(trait):Site,
             prior=list(R=list(V=R_scale,nu=3),
                  G=list(G1=list(V=diag(Run_scale),nu=3),
                  G2=list(V=diag(Site_scale),nu=3))),
             nitt = 130000,
             burnin = 30000,
             thin=10)
```

For each variance component, provide the scale matric and degrees of freedom for the prior distribution (inverse-Wishart.

Objective: Simulate the Complete Process
How ?

2. Posterior Predictions

If the measured CQAs remain the same during the whole process (e.g. API concentration), the predictions at step N could be written (random effects are omitted for simplicity):

$$p(\tilde{y} \mid \mathbf{y}) = \int_{\boldsymbol{\theta}} p(\tilde{y} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{y})\, d\boldsymbol{\theta}$$

$$Y^* = \underbrace{\sum_{s=1}^{N} [X_s^* \mathrm{B}_s]}_{\textit{True value (model estimates)}} \quad + \quad \underbrace{\mathrm{E}_N}_{\textit{Measurement uncertainty at step N}}$$

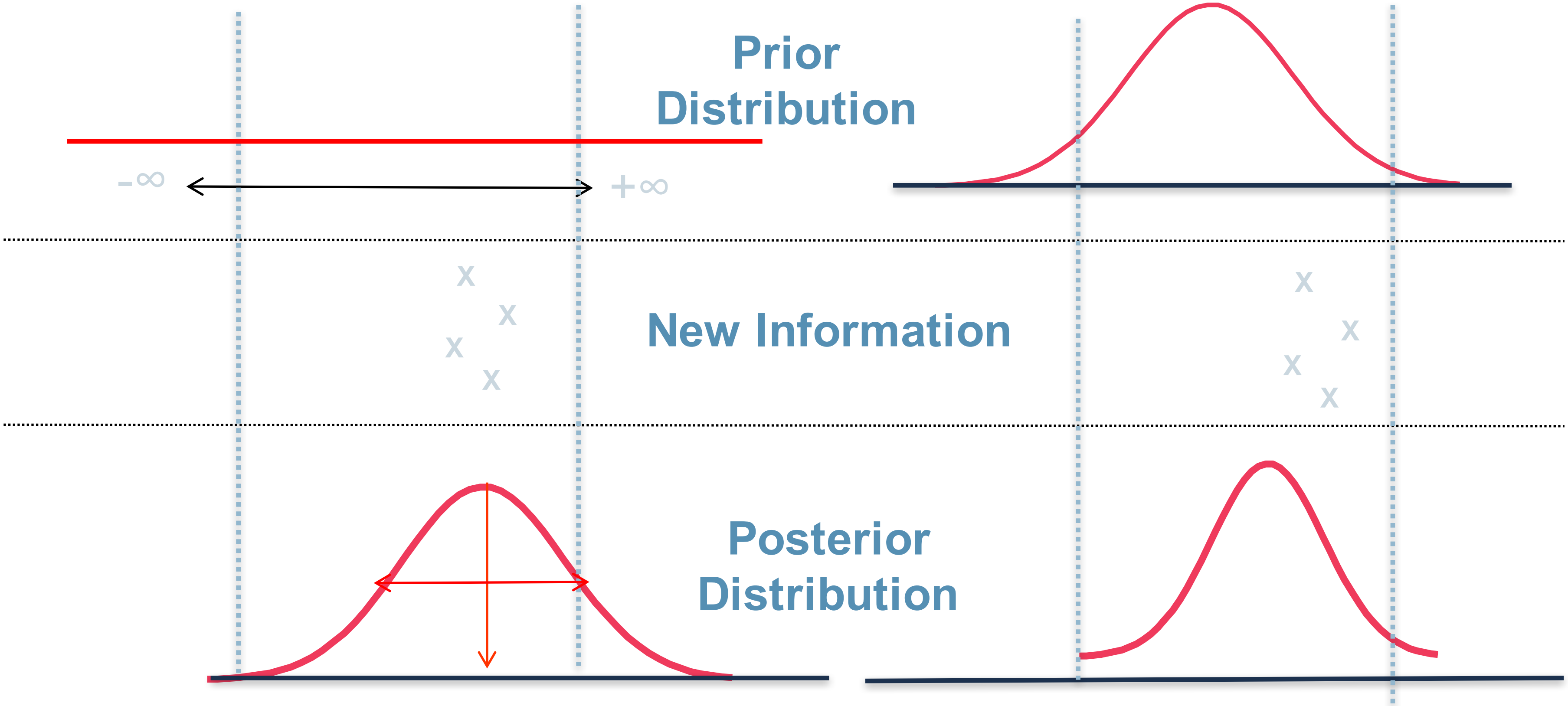
Where:

- Y^* is the new matrix of responses
- X_s^* is the new matrix of factors at step s
- $\mathrm{B}_s$is the matrix containing the posterior distributions of regression parameters at step s
- E_N is the matrix of measurement error at step *N*

Use of prior knowledge

- **Use your prior knowledge!** For example: the distribution of measurement error at each step should be known.
- When several CQAs are to be modeled together, the variance-covariance matrix might be tricky to model. **Use informative prior distributions**.
- Conjugate prior for scale parameters is the Inverse Wishart, defined by two parameters: degrees of freedom, and scale matrix.
- It’s sometimes not easy to interpret the Wishart distribution. A possible solution is the use of Correlation matrix instead of Covariance matrix (but need to go from ”simple” MCMCglmm to in-house implementation of e.g. a Stan sampler)

Use of prior knowledge



Prior Distribution

New Information

Posterior Distribution

Based on a point estimates

Based on a distribution

Frequentist

Bayesian

Objective: Simulate the Complete Process How ?

2. Posterior Predictions (continued)

$$p(\tilde{y} \mid \mathbf{y}) = \int_{\boldsymbol{\theta}} p(\tilde{y} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{y})\, d\boldsymbol{\theta}$$

From posterior chains of parameters, draw the joint posterior prediction of CQAs (Y) at the target operating conditions (X).

Sometimes, Process Parameters (PPs) are not 100% controlled. It is possible to randomly draw PPs across a specific range.

At every step, **replace the posterior chain of intercept with the predictive distribution of the previous step** (input distribution)**.**

The measurement error should only be included when “observing” the data (see later)

At every step, obtain marginal and joint probabilities to meet specifications, when they exist

Predictions from MCMCglmm model

```
### Simulation of uncontrolled process parameters (CENTERED)
new.data = data.frame(
  time = seq(-6,6,1),
  factor1 = rnorm(1e4, 0, 5)
  )
modmat <- model.matrix(~time*factor1, new.data)

### Predictions from Posterior Chains
### MCMCglmm model is m
Y = matrix(nrow = 1e4, ncol = 2)
for(i in 1:nrow(Y)){
  Beta <- t(matrix(m$Sol[i,], 2, 4))

  E_Run <- mvrnorm(1,rep(0,2), Sigma = diag(m$VCV[i,1:2]))
  E_Site <- mvrnorm(1,rep(0,2), Sigma = diag(m$VCV[i,3:4]))
  E_Residuals <- mvrnorm(1,rep(0,2), Sigma = matrix(m$VCV[i,5:8], 2, 2))

  Y[i,] <- modmat[i,] %*% Beta + E_Run + E_Site + E_Residuals
}
```

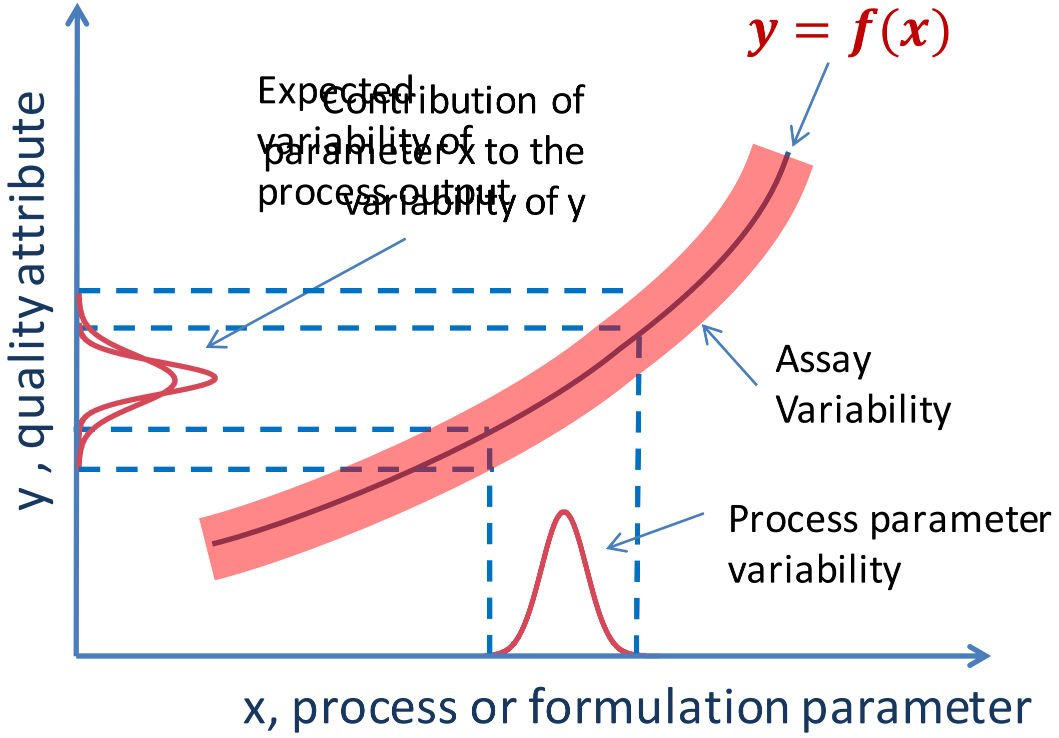
Sensitivity / robustness of Process and Formulation Parameters

Process characterization studies can be used to simulate the distribution of Quality Attributes across ranges of the CPPs

Determine $\boldsymbol{y = f(x)}$ from process or formulation studies

Utilizes normal variation in process parameters

Platform knowledge, manufacturing facility, and equipment may inform about parameter variability

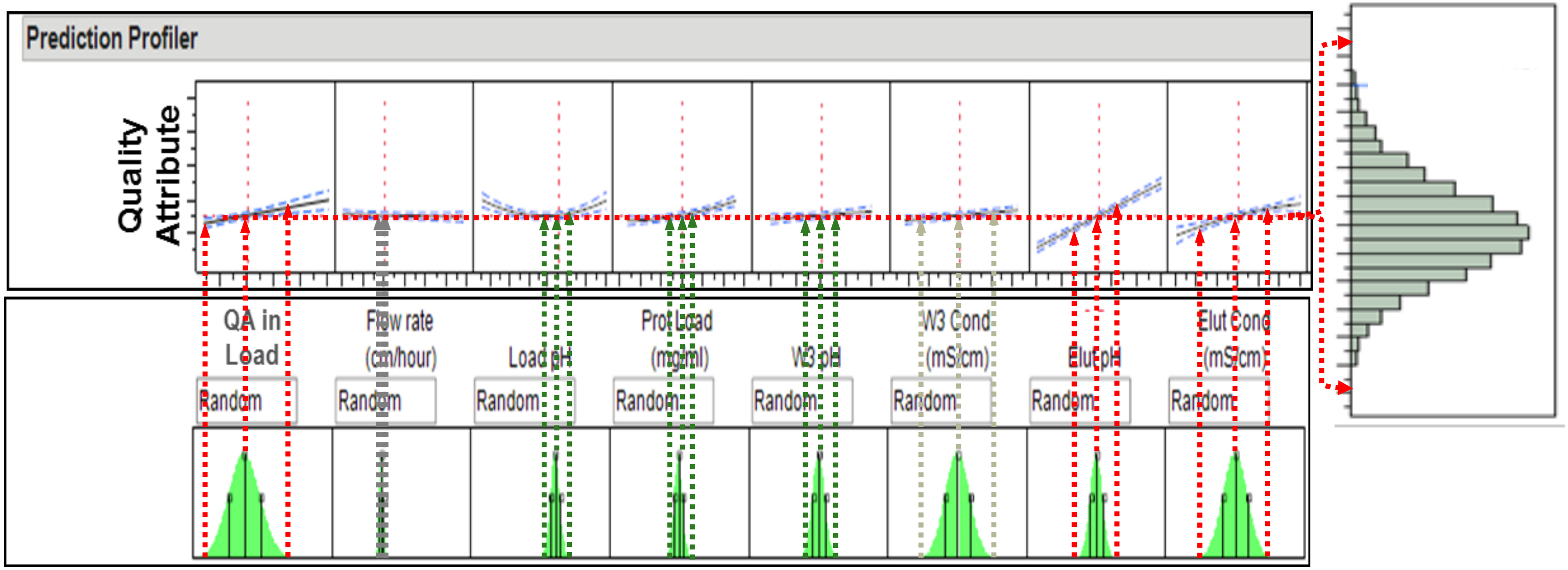

$y = f(x)$
Expected variability of process output
Contribution of parameter x to the variability of y
Assay Variability
Process parameter variability
y , quality attribute
x, process or formulation parameter

From **Miro Quesada et al**., Process and Shelf-Life Models to Complement Justification of Specifications, PDA MAb Workshop, September 22nd, 2015, Berlin, Germany

Identify Primary Source of Variability and Estimate prediction and P(OOS)

Parameters that impact CQAs and have a sufficiently wide distribution are likely sources of variability

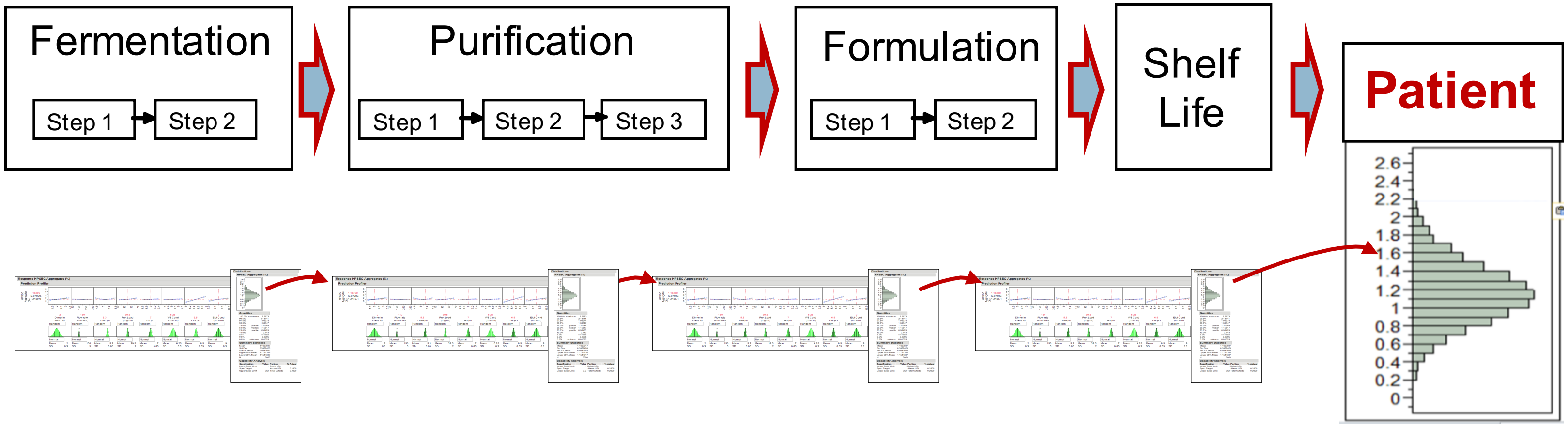
- ↑ No impact and very narrow distribution
- ↑ Some impact but narrow distribution
- ↑ Meaningful sources of variability
- ↑ Some Impact and wider distribution



From **Miro Quesada et al.**, Process and Shelf-Life Models to Complement Justification of Specifications, PDA MAb Workshop, September 22nd, 2015, Berlin, Germany

Combining Process Steps and Stability

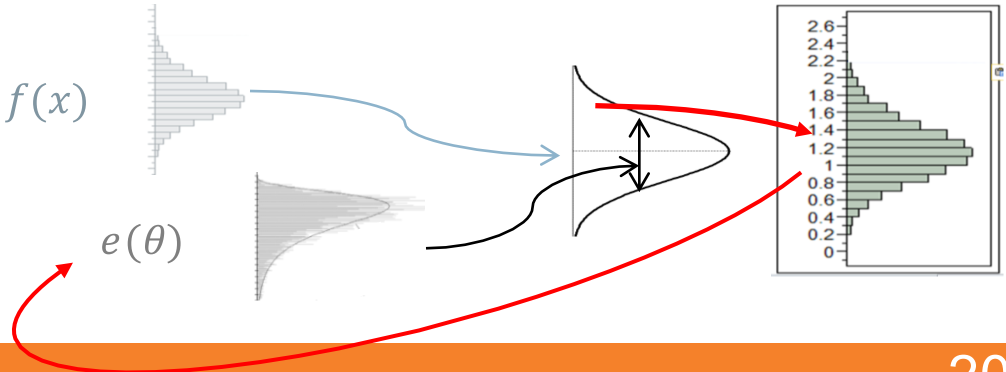
Output from previous unit operation used as input to the next: $y_{t+1} = f(x, y_t)$



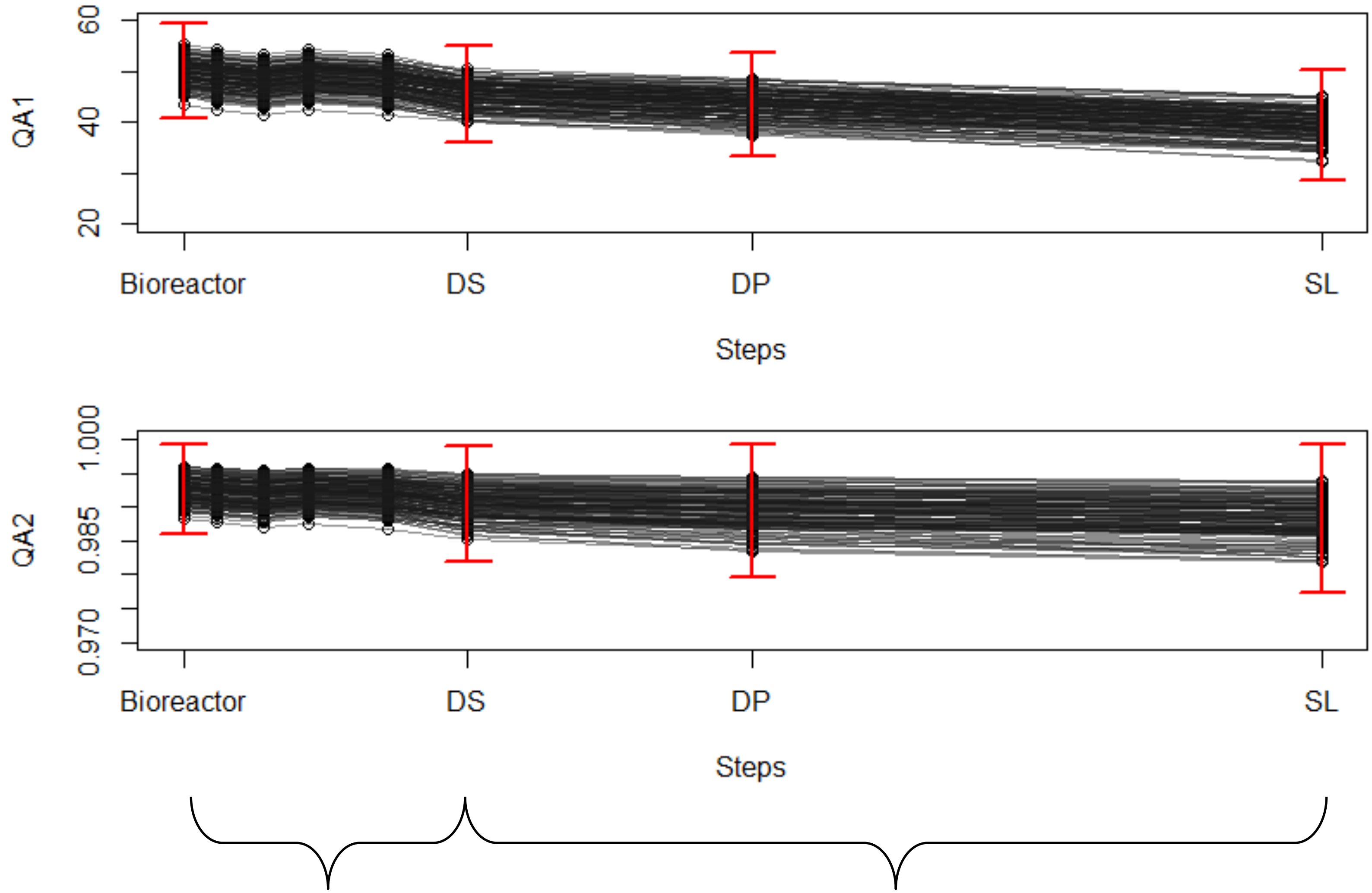
Statistical uncertainty – Bayesian analysis:

$$y_{t+1} = f(x, y_t) + e(process\ parameters) + s$$

From **Miro Quesada et al.**, Process and Shelf-Life Models to Complement Justification of Specifications, PDA MAb Workshop, September 22nd, 2015, Berlin, Germany



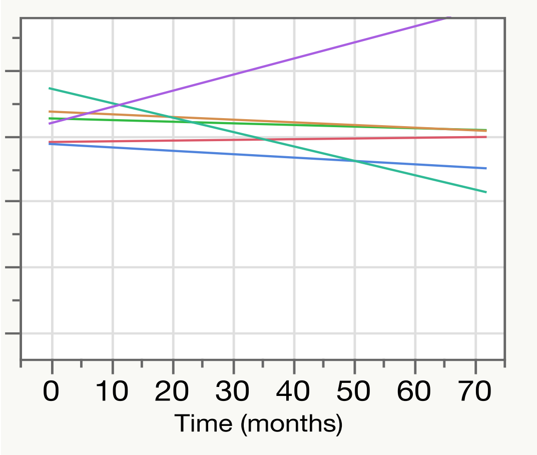
Outcome

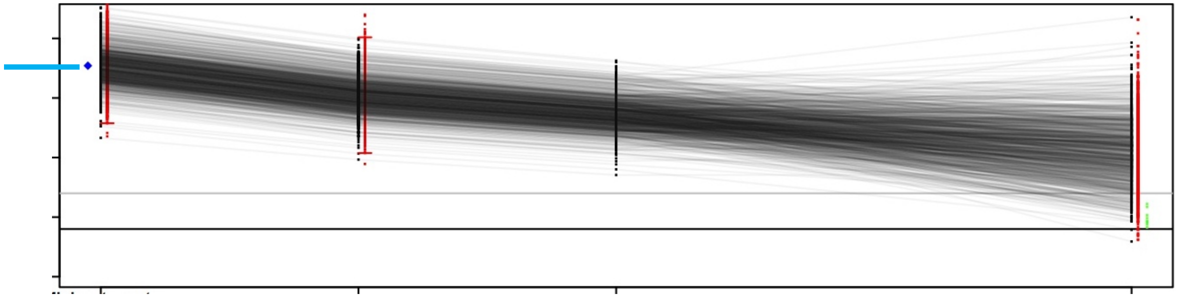


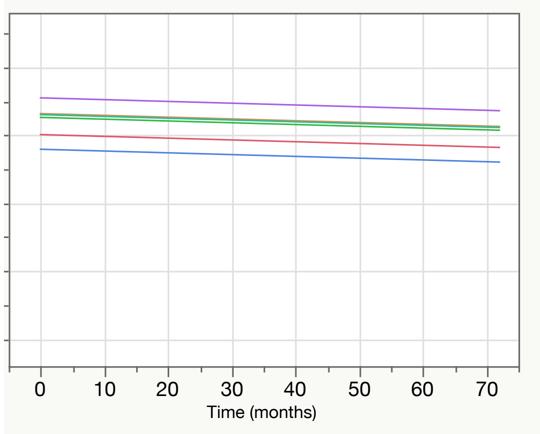
Manufacturing Process **Stability (holding times, Shelf-life)**

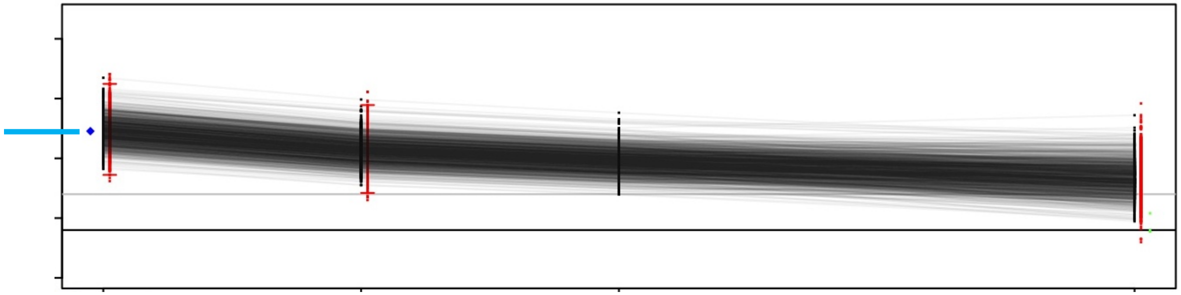
Impact of changes of the process

- Example : a change of formulation improving long term stability and batch-to-batch variability



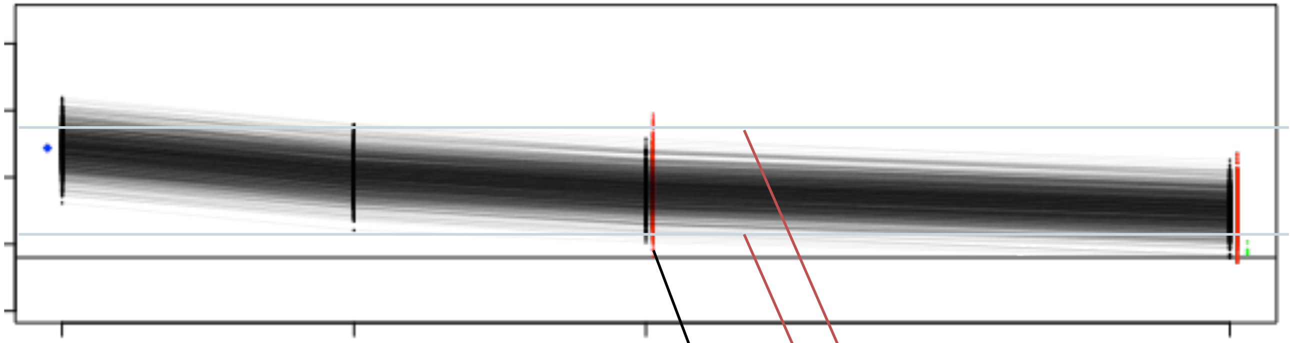


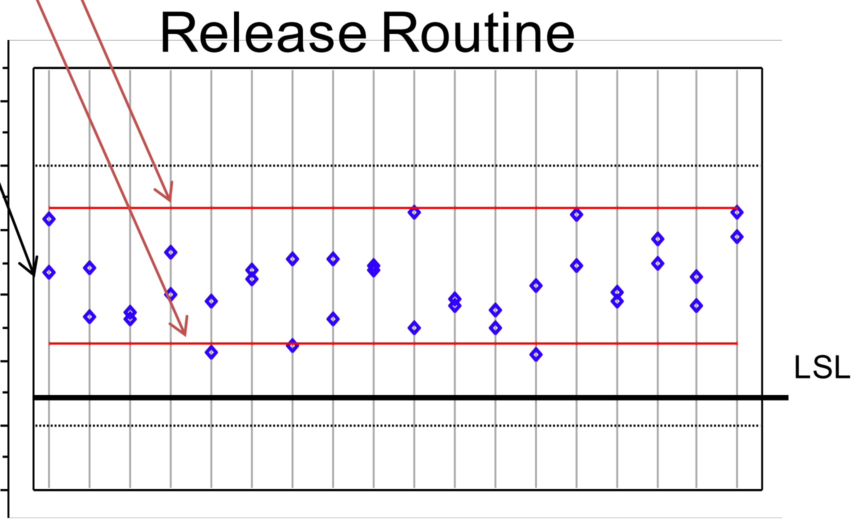




Control strategy

- Raise preliminary but appropriate out-of-control, alert, and batch rejection at release





It allows controlling the risks and keep the quality constant over time.

You maintain your initial claim and monitor it with appropriate levels of risk.

Take Home Messages

- Bayesian statistics allow the simulation of a sample from the bioreactor to the Drug Product shelf life.
- SPC generally requires a rather large amount of batches to derive control limits and/or estimate the capability of a process. Predictions of the complete process allow estimating the P(OOS) at each step, with far fewer observations, and using historical data

  Hence, Bayesian predictive intervals can also be used as preliminary control limits, defining a plausible control strategy, sooner during the development !
- Prior distributions, when informative, can result in a massive gain in degrees of freedom, directly impacting predictive uncertainty!

- **Prediction** is the key…

Acknowledgement

- Guillermo Miro-Quesada
- Timothy Schofield
- Harry Yang

Thank You !

But also

- Jenny Main
- Tara Scherder
- Bruno Boulanger
- Katherine Giacoletti

- Pol Adriaansen
- Olga Labovitiadi
- Piet Hoogkamer
- Sven-Daniel Schmitz