# F: TMB and MCMC

Fish 559; Day 3: 09h30-10h30

559

# Basics of a Bayesian Assessment

- Bayes Theorem:

$$p(\underline{\theta} \mid D) = \frac{L(D \mid \underline{\theta})\, p(\underline{\theta})}{\int L(D \mid \underline{\theta})\, p(\underline{\theta})\, d\underline{\theta}}$$

- We need to specify the prior distribution, $p(\underline{\theta})$, in order to apply Bayes theorem.
- The objective is to represent the posterior by means of a (large) number of vectors of parameters.

# Objectives for MCMC

- MCMC is a way to construct a (joint) Bayesian posterior for the parameters of a population dynamics model.
- The basic idea is to set up a (long) "chain" that starts at a pre-specified vector and that then traverses the posterior distribution.
- The sample from the posterior distribution is then every n'th element in the chain (ignoring the first "few" elements of the chain).

# Overview of the MCMC algorithm

1. Select an "initial" parameter vector (often the mode of the posterior) and compute its posterior density (the product of the likelihood and the prior).
2. Generate a "new" parameter vector based on the current one (using a "jump" function) and compute its posterior density.
3. Replace the current parameter vector by the new parameter vector with probability equal to the ratio of the new to the current posterior density.
4. Output the current parameter vector.
5. Repeat steps 2)- 4) many times!

Steps 2-4 are referred to as a cycle.

# A Little More Detail

A) Set $i$ to 0.

B) Set $y$ to $x_i$.

C) Apply the "jump function" to $y$ to generate a new vector $z$ (may only differ in one element from $y$) and compute $f(z)$.

D) If $f(z) > f(y)$, set $i = i+1$, $x_i = z$, go to B).

E) Generate $w \sim U[0,1]$. If $w < f(z)/f(y)$ , set $i = i+1$, $x_i = z$, go to B).

F) Set $i = i+1$, $x_i = y$ and go to B).

# The Jump Function

- The "jump" function should be chosen to optimize performance (but is usually selected for computational convenience).
- It should be possible to reach all possible parameter vectors "eventually" by applying the "jump" function long enough.
- The jump function depends on which MCMC algorithm you select. I have used:
  - A naive "Metropolis" algorithm
  - The No-Uturn sampler that can be found in the adnuts package. We will discuss this sampler in Lecture G.

# Example-I

- Compute posterior distributions for the parameters of a logistic regression.
  - Formulate the problem as maximum likelihood estimation.
  - Select prior distributions.
  - Choose a jump function.

# Example-II

- The likelihood formulation of the problem.

$$-\ell\text{n}L = n.\ell\text{n}\sigma + \frac{1}{2\sigma^2}\sum_i \left(\text{logit}(y_i) - \text{logit}\left(\left[1 + \exp(-\ell n 19\frac{x_i - x_{50}}{x_{95} - x_{50}})\right]^{-1}\right)\right)$$

- The prior: uniform on the logarithms of $x_{50}$, $x_{95}$ and $\sigma$.
- The jump function: multivariate normal with mean zero and variance-covariance matrix from a TMB fit.
- Hint: You will need the mvtnorm library to run the mcmc function.
- The code is in LectF1.R.

# Example-III
## (R Implementation)

```
f1 <- function(x)
{
 A50 <- exp(x[1])
 A95 <- exp(x[2])
 Sigma <- exp(x[3])
 pred <- MakePred(co$Length,A50,A95)
 pred <- log(pred/(1-pred))
 SS <- (pred-co$Prob2)^2
 SS <- sum(SS)
 LogLike <- length(pred)*log(Sigma) +
sum(SS)/(2.0*Sigma^2)
 return(LogLike)
}
```

Calculate the likelihood function

```
MakePred<-function(Length,A50,A95)
{
 pred <- 1/(1+exp(-1*log(19)*(Length-
A50)/(A95-A50)))
 return(pred)
}
```

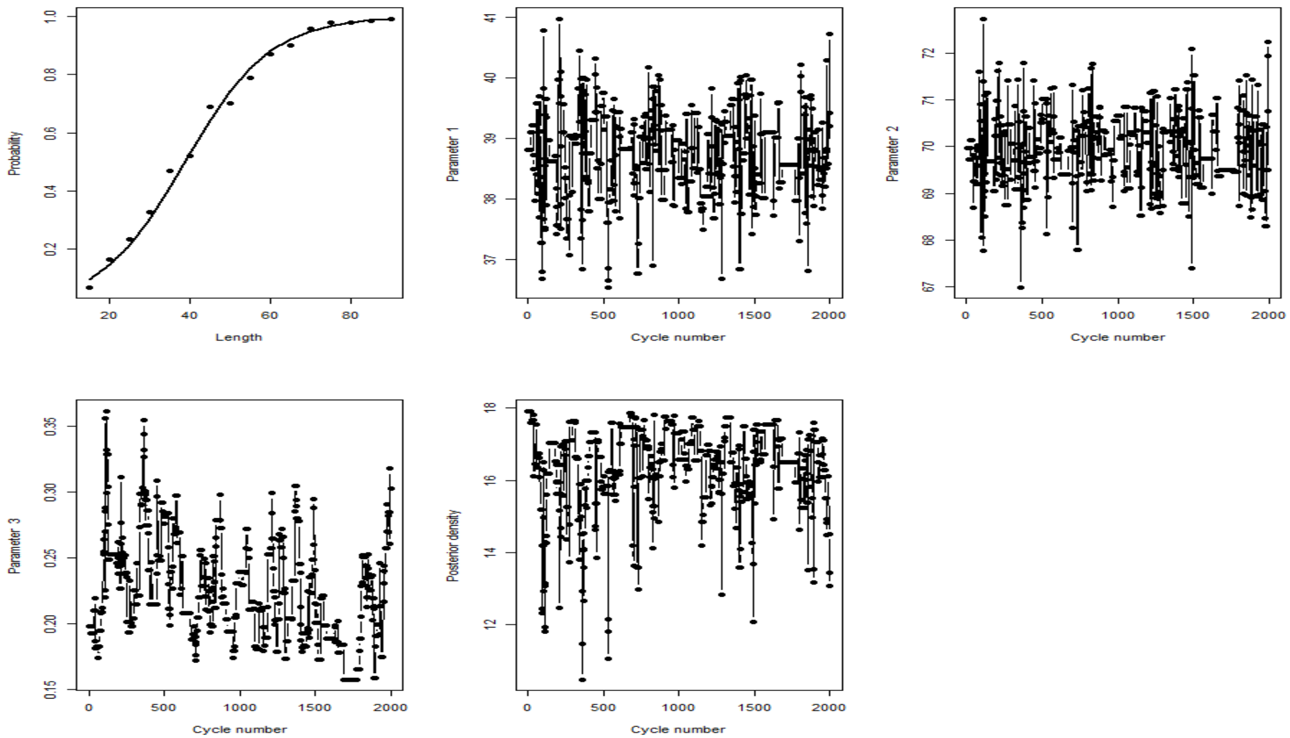
Vector of predicted values

# Example-IV (R Implementation)

```
DoMCMC<-
function(Xinit,Ndim,sd,cor,Nsim=1000,
Nburn=0,Nthin=1)
{
 FileN <- "lect18.txt"
 TheData<-scan(file=FileN,
what=list(Length=0,Prob=0))
 co <- NULL
 co$Length <- TheData$Length
 co$Prob1 <- TheData$Prob
 co$Prob2 <- log(TheData$Prob/(1-
TheData$Prob))
 environment (f1) =environment()
```

```
Xcurr <- Xinit
Fcurr <- -1*f1(Xcurr)
Outs <- matrix(0,nrow=(Nsim-Nburn),ncol=(Ndim+1))
Ipnt <- 0; Icnt <- 0
for (Isim in 1:Nsim)
 {
  Xnext <- rmvnorm(1, Xcurr, cov=cor, sd, rho, Ndim)
  Fnext <- -1*f1(Xnext)
  Rand1 <- log(runif(1,0,1))
  if (Fnext > Fcurr+Rand1)
   {Fcurr <- Fnext; Xcurr <- Xnext }
  if (Isim %% Nthin == 0)
   {
    Ipnt <- Ipnt + 1
    if (Ipnt > Nburn)
     { Icnt <- Icnt + 1; Outs[Icnt,] <- c(Xcurr,Fcurr); }
   }
 }
}
```

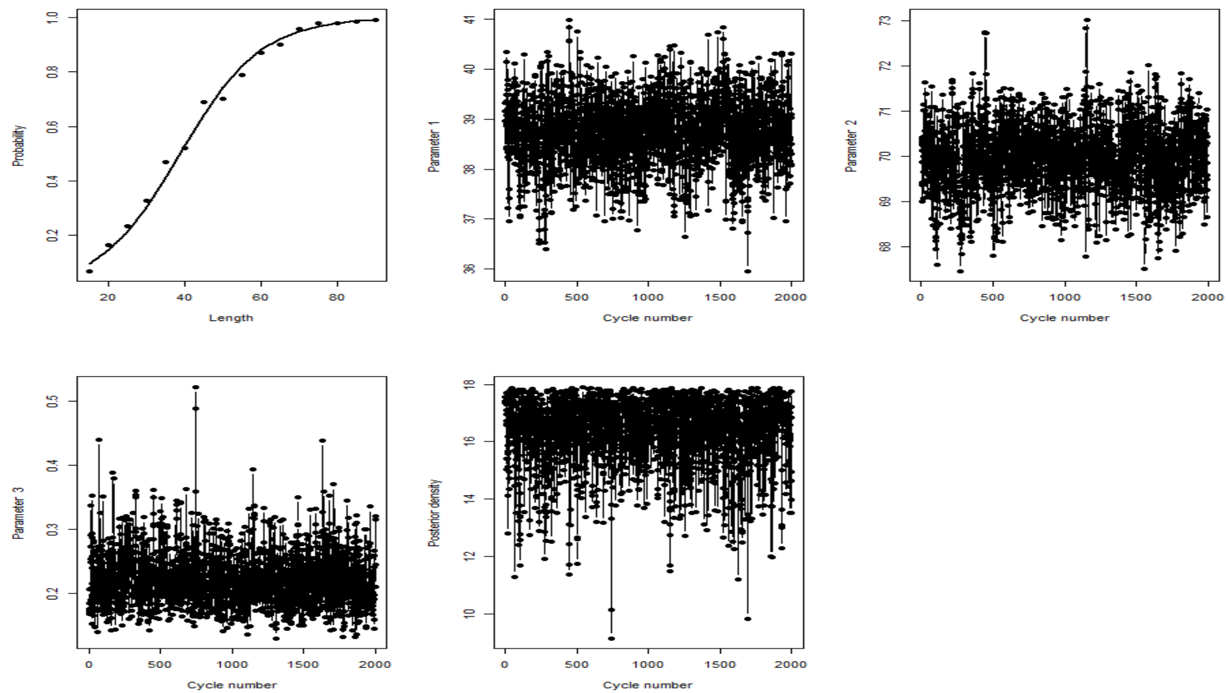
# Example-V (Results-silly jump function)




**sd <- c(0.1,0.1,0.1); cor <- matrix(c(1,0,0,0,1,0,0,0,1), ncol=3,nrow=3)**

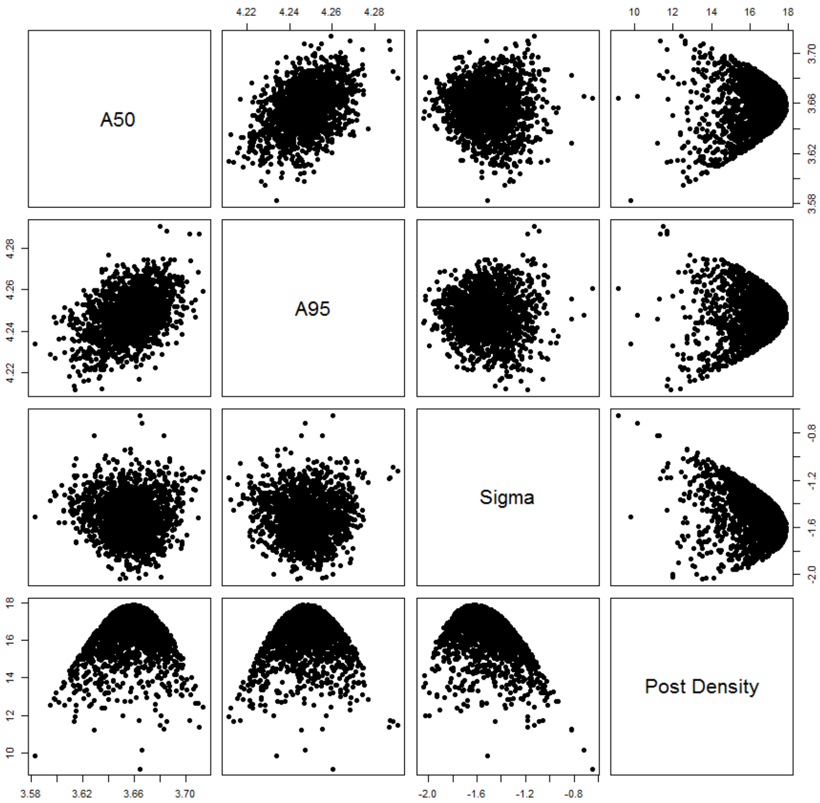
559

# Example-VI (Results-Informative jump function)




```
sd <- c(0.015702,0.0093949,0.17678);
cor <- matrix(c(1,0.3769,0,0.3769,1,0,0,0,1),ncol=3,nrow=3)
```

# Example-VII (Results-Posterior correlations)



559

# Example-IX

This problem can be done in one step because one can call TMB functions from R (see LectF2.R).

In general, the steps should be:

- Fit the model (maximum likelihood)
- Obtain the variance-covariance matrix (hessian=T)
- Call MCMC
- View MCMC diagnostics.

# "Burn in" and "thinning"

- The impact of the initial parameter vector and the Markov nature of the algorithm are reduced by having a "burn in" period and by "thinning" the chain.
- To get *x* samples from the posterior:
  - Allow for a "burn-in" period (5-50% of the total chain length) to allow the algorithm to "set itself up". The results of this period are discarded.
  - "Thin" the chain by taking every *n*'th value in the chain.

# Tricks of the (MCMC) trade - I

- The performance of MCMC deteriorates as the number of parameters increases. Therefore, integrate out any nuisance parameters analytically.
  - Examples (Walters and Ludwig, 1994).
    - The catchability coefficients;
    - The observation error standard deviations.

Walters, C.J. and D. Ludwig. *Can. J. Fish. Aquat Sci*. 51: 713-722

559

# Tricks of the (MCMC) trade - II

- The performance of MCMC deteriorates if the parameters are highly correlated (often true for age-structured models).
- Use the .COR file to identify highly-correlated parameters ($\rho \geq 0.8$).
- Re-parameterize the model to reduce correlations:

Use

$$S_a = \left(1 + \exp[-\frac{(a - a_{50})}{\Delta}]\right)^{-1}$$

and not

$$S_a = \left(1 + \exp[-\frac{(a - a_{50})}{(a_{95} - a_{50})}]\right)^{-1}$$