

**Supplemental Materials S1.** JAGS script that implements the Bayesian probabilistic modeling platform for the developed individual-tree growth and yield model.

```
model{
  # per ha expansion factor for individual trees on projection plots
  expansion <- 4.942

  # initial derived variables
  # var k is projection year, k=1 is initial year before projection
  for( k in 1:1 ) {
    AGE.SI <- 25
    SI[k] ~ dnorm(mu.HD[k], tau.HD.proj)
    mu.HD[k] <- pow(AGE.SI, 2) / (parms.hd[1] + AGE.SI *
      (TAGE[k] / HD[k] - parms.hd[2] * TAGE[k] - parms.hd[1] / TAGE[k] + parms.hd[2] *
        AGE.SI))
    SI.projection <- max(1, abs(AGE.SI - TAGE[k]))
    tau.HD.proj <- (1 / SI.projection) * tau[7]
  }

  # initial derived variables for individual trees (var i is individual-tree)
  for( i in 1:ntrees ) {
    for( k in 1:1 ) {

      # set initial survival to unity
      I.surv[i , k] <- 1

      # compute ind-tree basal area
      BAI[i , k] <- (3.1415 / 40000) * pow(D[i , k], 2) * expansion

      # likelihood function for estimating missing heights with measured H and D data used
      # to estimate parameters (prior distributions given later in the script)
      ln.H[i , k] ~ dnorm(mu.H[i , k], tau.H)
      mu.H[i , k] <- parms.b[1] + parms.b[2] * log(D[i , k])
      H[i , k] <- exp(ln.H[i , k])

      # likelihood function for estimating missing height to crown base with measured HCB
      # and D data used to estimate parameters (prior distributions given later in the script)
      ln.HCB[i , k] ~ dnorm(mu.HCB[i , k], tau.HCB)
      mu.HCB[i , k] <- parms.c[1] + parms.c[2] * log(D[i , k])
      HCB.x[i , k] <- exp(ln.HCB[i , k])
      # constrain HCB < H
      HCB[i , k] <- min(HCB.x[i , k], H[i , k] - 1)

      # compute maximum crown width
      MCW[i , k] <- 1.4081 + 0.22111 * D[i , k] - 5.3438E-4 * pow(D[i , k], 2)

      # compute maximum crown area
      MCA[i , k] <- I.surv[i , k] * expansion * 100 * pow(MCW[i , k], 2) * 3.1415 / 40000
    }
  }
}
```

```

    # compute crown length and crown ratio
    CL[i , k] <- H[i , k] - HCB[i , k]
    CR[i , k] <- CL[i , k] / H[i , k]
  }
}

```

```

# compute initial year, plot summation variables

```

```

for( k in 1:1 ) {
  BA[k] <- sum(sub.ba[ , k])
  TPH[k] <- sum(sub.tph[ , k])
  DQ[k] <- 100 * sqrt(4 * BA[k] / 3.1415 * TPH[k])
  CCF[k] <- sum(MCA[ , k])
  for( i in 1:ntrees ) {
    sub.ba[i , k] <- BAI[i , k]
    sub.tph[i , k] <- expansion
  }
}

```

```

### start full projection

```

```

for( k in 2:nyears ) {

```

```

  # increment projection year

```

```

  TAGE[k] <- TAGE[1] + k - 1

```

```

  # likelihood function for dominant height projection (Note that only HD[1] was data)

```

```

  HD[k] ~ dnorm(HD.mu[k], tau[7])
  HD.mu[k] <- pow(TAGE[k], 2) / (parms.hd[1] + TAGE[k] *
    (TAGE[k - 1] / HD[k - 1] - parms.hd[2] * TAGE[k - 1] - parms.hd[1] / TAGE[k - 1] +
    parms.hd[2] * TAGE[k]))
}

```

```

# tree-level projections

```

```

for( i in 1:ntrees ) {

```

```

  for( k in 2:nyears ) {

```

```

    # compute growth effective age (Note that only H[i, 1] was data)

```

```

    GEA[i , k] <- ( -1 ) * z[i , k] + sqrt(H[i , k - 1] * parms.hd[1] / (1 - H[i , k - 1] * parms.hd[2]) +
      pow(z[i , k], 2))

```

```

    z[i , k] <- (H[i , k - 1] * parms.hd[1] / AGE.SI + H[i , k - 1] * parms.hd[2] * AGE.SI -
      H[i , k - 1] * AGE.SI / SI[1]) / 2 * (1 - H[i , k - 1] * parms.hd[2])

```

```

    # compute potential height growth

```

```

    mu.PHG[i , k] <- pow(GEA[i , k] + 1, 2) / (parms.hd[1] + (GEA[i , k] + 1) *
      (GEA[i , k] / H[i , k - 1] - parms.hd[2] * GEA[i , k] - parms.hd[1] / GEA[i , k] +
      parms.hd[2] * (GEA[i , k] + 1))) - H[i , k - 1]

```

```

    # compute height modifier

```

```

    HMOD[i , k] <- pow(H[i , k - 1] / HD[k - 1], parms.f[1] + install[3] + plot[3]) *
      (1 - parms.f[2] * pow(CCF[k - 1] / 600, parms.f[3]))

```

**# likelihood function for height increment**

```
dHG[i , k] ~ dnorm(mu.HG[i , k], tau.HG.w[i , k])
mu.HG[i , k] <- mu.PHG[i , k] * HMOD[i , k]
tau.HG.w[i , k] <- (1 / pow(abs(H[i , k - 1]), 2 * 0.3443105)) * tau[3]
```

**# constrain height increment >= 0**

```
dHG.pos[i , k] <- max(dHG[i , k], 0)
H[i , k] <- dHG.pos[i , k] * I.surv[i , k - 1] + H[i , k - 1]
```

**# likelihood function for height to crown base increment**

```
dHCB[i , k] ~ dnorm(mu.dHCB[i , k], tau.dHCB.w[i , k])
mu.dHCB[i , k] <- (CL[i , k - 1] + mu.HG[i , k]) / (1 + exp(parms.dc[1] + install[5] + plot[5] +
  parms.dc[2] * log(CR[i , k - 1]) + parms.dc[3] * CR[i , k - 1] + parms.dc[4] * (GEA[i , k] -
  1) + parms.dc[5] * log(BA[k - 1]) + parms.dc[6] * (CR[i , k - 1] / BA[k - 1])))
tau.dHCB.w[i , k] <- (1 / pow(abs(CL[i , k - 1]), 2 * 0.9103295)) * tau[5]
```

**# constrain crown base increment >= 0**

```
dHCB.pos[i , k] <- max(dHCB[i , k], 0)
HCB.a[i , k] <- HCB[i , k - 1] + dHCB.pos[i , k]
```

**# constrain crown base < height**

```
HCB[i , k] <- min(HCB.a[i , k], H[i , k] - 1)
```

**# compute maximum crown width**

```
MCW[i , k] <- 1.4081 + 0.22111 * D[i , k] - 5.3438E-4 * pow(D[i , k], 2)
```

**# compute maximum crown area**

```
MCA[i , k] <- I.surv[i , k] * expansion * 100 * pow(MCW[i , k], 2) * 3.1415 / 40000
```

**# compute crown length and crown ratio**

```
CL[i , k] <- H[i , k] - HCB[i , k]
CR[i , k] <- CL[i , k] / H[i , k]
```

**# likelihood function for diameter increment**

```
DG.u[i , k] ~ dnorm(mu.DG[i , k], tau.DG.w[i , k])
mu.DG[i , k] <- exp(parms.e[1] + install[4] + plot[4] + parms.e[2] * D[i , k - 1] + parms.e[3] *
  pow(D[i , k - 1], 2) + parms.e[4] * log((CR[i , k - 1] + 0.2) / 1.2) + parms.e[5] * log(SI[1] -
  1.37) + parms.e[6] * log(D[i , k - 1] / DQ[k - 1]) + parms.e[7] * sqrt(BA[k - 1]))
tau.DG.w[i , k] <- (1 / pow(abs(D[i , k - 1]), 2 * 0.04456)) * tau[4]
```

**# constrain D increment >= 0, and set D constant if mortality event**

```
D[i , k] <- D[i , k - 1] + max(0, DG.u[i , k]) * I.surv[i , k - 1]
```

**# compute ind-tree basal area**

```
BAI[i , k] <- (3.1415 / 40000) * pow(D[i , k], 2) * expansion
```

**# likelihood function for survival**

```
Surv[i , k] ~ dbern(p.s.x[i , k])
p.s.x[i , k] <- max(0, min(p.s[i , k], 1))
```

```

logit(p.s[i , k]) <- parms.g[1] + parms.g[2] * D[i , k - 1] + parms.g[4] * SI[1] +
  parms.g[5] * (1 - D[i , k - 1] / DQ[k - 1]) + parms.g[6] * BA[k - 1]

```

```

# update survival indicator variable (0,1) and constrain across projection years

```

```

I.surv[i , k] <- min(I.surv[i , k - 1], Surv[i , k])
}

```

```

}

```

```

# compute plot summation vars

```

```

for( k in 2 : nyears ) {
  BA[k] <- sum(sub.ba[ , k])
  TPH[k] <- sum(sub.tph[ , k])
  DQ[k] <- 100 * sqrt(4 * BA[k] / 3.1415 * TPH[k])
  CCF[k] <- sum(MCA[ , k])
  for( i in 1 : ntrees ) {
    sub.ba[i , k] <- BAI[i , k] * I.surv[i , k]
    sub.tph[i , k] <- expansion * I.surv[i , k]
  }
}

```

```

# Begin specifying parameter distributional forms for prior distributions for missing H and HCB

```

```

for( i in 1:2 ) {
  parms.b[i] ~ dnorm(0, 0.0001)
  parms.c[i] ~ dnorm(0, 0.0001)
}

```

```

# precision prior for H, using a partially informative prior

```

```

tau.H <- 1 / pow(sigma.H, 2)
sigma.H ~ dunif(0,20)

```

```

# precision prior for HCB, using a partially informative prior

```

```

tau.HCB <- 1 / pow(sigma.HCB, 2)
sigma.HCB ~ dunif(0, 20)

```

```

### MVN parameters

```

```

# height to crown base parameters

```

```

for( i in 1 : 6 ) {
  for( j in 1 : 6 ) {
    Var.dC[i , j] <- rho.dC[i , j] * sigma.dC[i] * sigma.dC[j]
  }
}
parms.dc[1:6] ~ dmnorm(parms.dC[1:6], tau.dC[1:6 , 1:6])
tau.dC[1:6 , 1:6] <- inverse(Var.dC[1:6 , 1:6])

```

```

# height growth modifier parameters

```

```

for( i in 1 : 3 ) {
  for( j in 1 : 3 ) {
    Var.F[i , j] <- rho.F[i , j] * sigma.F[i] * sigma.F[j]
  }
}
}

```

```
parms.f[1:3] ~ dmnorm(parms.F[1:3], tau.F[1:3 , 1:3])
tau.F[1:3 , 1:3] <- inverse(Var.F[1:3 , 1:3])
```

#### **# diameter increment parameters**

```
for( i in 1 : 7 ) {
  for( j in 1 : 7 ) {
    Var.E[i , j] <- rho.E[i , j] * sigma.E[i] * sigma.E[j]
  }
}
parms.e[1:7] ~ dmnorm(parms.E[1:7], tau.E[1:7 , 1:7])
tau.E[1:7 , 1:7] <- inverse(Var.E[1:7 , 1:7])
```

#### **# survival parameters**

```
for( i in 1 : 6 ) {
  for( j in 1 : 6 ) {
    Var.G[i , j] <- rho.G[i , j] * sigma.G[i] * sigma.G[j]
  }
}
parms.g[1:6] ~ dmnorm(parms.G[1:6], tau.G[1:6 , 1:6])
tau.G[1:6 , 1:6] <- inverse(Var.G[1:6 , 1:6])
```

#### **# dominant height growth parameters**

```
for( i in 1 : 2 ) {
  for( j in 1 : 2 ) {
    Var.hD[i , j] <- rho.hD[i , j] * sigma.hD[i] * sigma.hD[j]
  }
}
parms.hD[1:2] ~ dmnorm(parms.hD[1:2], tau.hD[1:2 , 1:2])
tau.hD[1:2 , 1:2] <- inverse(Var.hD[1:2 , 1:2])
```

#### **# installation and plot random effects**

```
for ( i in 3:5 ) {
  install[i] ~ dnorm(0, tau.i[i])
  plot[i] ~ dnorm(0, tau.p[i])
}
for( i in 1 : 7 ) {
  tau.i[i] <- 1 / pow(sigma.i[i], 2)
  tau.p[i] <- 1 / pow(sigma.p[i], 2)
}
```

#### **# residual errors**

```
for( i in 1 : 7 ) {
  tau[i] <- 1 / pow(sigma.e[i], 2)
}
```

#### **# model end**

```
}
```