```
// Code generated by the R package readsdr v0.2.0.9014
// See more info at github https://github.com/jandraor/readsdr
functions {
  vector X_model(real time, vector y, array[] real params) {
    vector[12] dydt;
    real ER;
    real RR;
    real HR1;
    real HR2;
    real HR3;
    real RRH;
    real RRD;
    real Lambda;
    real TCI;
    real THI;
    real TDI;
    real TIH;
    real Checksum;
    real IR;
    ER = y[2]*0.5;
    RR = (1-params[1]*params[2])*y[3]*0.5;
    HR1 = params[1]*params[2]*y[3]*0.5;
    HR2 = y[5]/(10/3.0);
    HR3 = y[6]/(10/3.0);
    RRH = (1-params[3])*y[7]/(10/3);
    RRD = params[3]*y[7]/(10/3.0);
    Lambda = params[4]*y[3]/1e+05;
    TCI = params[1]*ER;
    THI = HR1;
    TDI = RRD;
    TIH = y[5]+y[6]+y[7];
    Checksum = y[1]+y[2]+y[3]+TIH+y[4]+y[8]+y[9];
    IR = y[1]*Lambda;
    dydt[1] = -IR;
    dydt[2] = IR-ER;
    dydt[3] = ER-RR-HR1;
    dydt[4] = RR;
    dydt[5] = HR1-HR2;
    dydt[6] = HR2-HR3;
    dydt[7] = HR3-RRH-RRD;
    dydt[8] = RRH;
    dydt[9] = RRD;
    dydt[10] = TCI;
    dydt[11] = THI;
    dydt[12] = TDI;
    return dydt;
  }
}
data {
  int<lower = 1> n_obs;
  array[n_obs] int C;
  array[n_obs] int H;
  array[n_obs] int D;
  array[n_obs] real ts;
  vector[12] x0;
}
```

```
parameters {
  real<lower = 0, upper = 1> CF;
  real<lower = 0, upper = 1> HF;
  real<lower = 0, upper = 1> DF;
  real<lower = 0> Beta_Param;
  real<lower = 0> inv_phi1;
  real<lower = 0> inv_phi2;
  real<lower = 0> inv_phi3;
}
transformed parameters{
  array[n_obs] vector[12] x; // Output from the ODE solver
  array[4] real params;
  array[n_obs] real delta_x_1;
  array[n_obs] real delta_x_2;
  array[n_obs] real delta_x_3;
  real phi1;
  real phi2;
  real phi3;
  phi1 = 1 / inv_phi1;
  phi2 = 1 / inv_phi2;
  phi3 = 1 / inv_phi3;
  params[1] = CF;
  params[2] = HF;
  params[3] = DF;
  params[4] = Beta_Param;
  x = ode_rk45(X_model, x0, 0, ts, params);
  delta_x_1[1] =  x[1, 10] - x0[10] + 1e-5;
  delta_x_2[1] =  x[1, 11] - x0[11] + 1e-5;
  delta_x_3[1] =  x[1, 12] - x0[12] + 1e-5;
  for (i in 1:n_obs-1) {
    delta_x_1[i + 1] = x[i + 1, 10] - x[i, 10] + 1e-5;
    delta_x_2[i + 1] = x[i + 1, 11] - x[i, 11] + 1e-5;
    delta_x_3[i + 1] = x[i + 1, 12] - x[i, 12] + 1e-5;
  }
}
model {
  CF ~ beta(2, 2);
  HF ~ beta(2, 2);
  DF ~ beta(2, 2);
  Beta_Param ~ lognormal(0, 1);
  inv_phi1 ~ exponential(5);
  inv_phi2 ~ exponential(5);
  inv_phi3 ~ exponential(5);
  C ~ neg_binomial_2(delta_x_1, phi1);
  H ~ neg_binomial_2(delta_x_2, phi2);
  D ~ neg_binomial_2(delta_x_3, phi3);
}
generated quantities {
  real log_lik;
  array[n_obs] int sim_C;
  array[n_obs] int sim_H;
  array[n_obs] int sim_D;
  log_lik = neg_binomial_2_lpmf(C | delta_x_1,
phi1)+neg_binomial_2_lpmf(H | delta_x_2, phi2)+neg_binomial_2_lpmf(D |
delta_x_3, phi3);
  sim_C = neg_binomial_2_rng(delta_x_1, phi1);
  sim_H = neg_binomial_2_rng(delta_x_2, phi2);
  sim_D = neg_binomial_2_rng(delta_x_3, phi3);
}
```