Supplementary Material

**Monte Carlo Simulations Suggest Current Chlortetracycline Drug-Residue Based Withdrawal Periods Would Not Control Antimicrobial Resistance Dissemination from Feedlot to Slaughterhouse**

**Casey L. Cazer\*, Lucas Ducrot, Victoriya V. Volkova, Yrjö T. Gröhn**

**\* Correspondence:** Casey L Cazer: clc248@cornell.edu

**Supplementary Figures**



**Figure S1**: Number of resistant, intermediate and susceptible *Escherichia coli* in the bovine large intestine with and without oral chlortetracycline treatment. The log10(*E. coli*) (Y-axis) are presented for 1,000 simulations of each treatment scenario (rows): the absence of chlortetracycline (CTC) treatment, CTC for the reduction of liver abscesses (ARLA), disease control (ADC) or disease treatment (ADT). Blue shaded band is the 95% non-parametric confidence interval of the median, black dashed lines are 25% and 75% percentiles and purple dashed lines are 5% and 95% percentiles of the overall distribution. Treatment durations are the same as in Figure 1.

**Supplementary Information**

**MATLAB Model Code**

%{

Abbreviations:

ARLA: antimicrobial reduction of liver abscesses

ADC: antimicrobial disease control

ADT: antimicrobial disease therapy

PD: pharmacodynamic

PK: pharmacokinetic

Pop: population

eco: E coli

li: large intestine

CTC: chlortetracycline

ss: steady state

prop: proportion

res: resistant

int: intermediate-resistant

susc: susceptible

SIR, xx: susceptible/intermediate/resistant

conc: concentration

st: stomach

upsi: upper small intestine

rtsi: rest of small intestine

pl: plasma

tis: tissue

amt: amount

cat: cattle

deg: degradation

tx: treatment

}%

INTRASTOCHA=0; %1 intra-individual stochasticity (square root terms)

%0=no stochastic;

%intra-individual stochasticity was not implemented in this model;

PKSTOCHA=1;

%1 interindividual stochasticity of PK parameters; %0 no stochastic;

POPSTOCHA=1;

%1 interindividual stochasticity of E. coli population parameters; %0 no

%stochastic;

PDSTOCHA=1;

%1 interindividual stochasticity of PD parameters; %0 no stochastic;

METHODE=1;

% 0=no CTC 1=ARLA; 2=ADC 3=ADT;

time=90\*24; %90 day simulation period;

dt=0.1;

am\_start=48; %when e coli population has established steady state

adt\_time=5\*24;

adc\_time=28\*24;

bw=300; %body weight;

%set up number of simulations;

if INTRASTOCHA==0 && PKSTOCHA==0 && POPSTOCHA==0 && PDSTOCHA==0

 n=1; %deterministic model

else

n=1000;

end

%making empty arrays to store stochastic simulations;

res\_eco\_li=zeros(time/dt,n);

int\_eco\_li=zeros(time/dt,n);

susc\_eco\_li=zeros(time/dt,n);

total\_eco=zeros(time/dt,n);

%number of e coli in large intestine;

total\_eco\_ss\_preCTC=zeros(1,n);

total\_eco\_ss\_CTC=zeros(1,n);

total\_eco\_ss\_postCTC=zeros(1,n);

prop\_res\_eco=zeros(time/dt,n);

prop\_int\_eco=zeros(time/dt,n);

prop\_susc\_eco=zeros(time/dt,n);

%proportion (percentage) of SIR out of total e coli;

prop\_susc\_ss\_postCTC=zeros(1,n);

prop\_int\_ss\_postCTC=zeros(1,n);

prop\_res\_ss\_postCTC=zeros(1,n);

%proportion of SIR out of total e coli at end of simulation;

t480\_susc\_eco\_li=zeros(1,n);

t480\_res\_eco\_li=zeros(1,n);

t480\_int\_eco\_li=zeros(1,n);

t480\_total\_eco=zeros(1,n);

%number of SIR ecoli at t=480 (start of CTC);

%finding prop\_xx\_eco value steady state before CTC;

prop\_res\_ss\_preCTC=zeros(1,n);

prop\_int\_ss\_preCTC=zeros(1,n);

prop\_susc\_ss\_preCTC=zeros(1,n);

prop\_res\_ss\_CTC=zeros(1,n);

prop\_int\_ss\_CTC=zeros(1,n);

prop\_susc\_ss\_CTC=zeros(1,n);

res\_pre\_CTC\_change=zeros(1,n);

int\_pre\_CTC\_change=zeros(1,n);

susc\_pre\_CTC\_change=zeros(1,n);

res\_pre\_post\_change=zeros(1,n);

int\_pre\_post\_change=zeros(1,n);

susc\_pre\_post\_change=zeros(1,n);

ctc\_li\_conc=zeros(time/dt,n);

ctc\_pl\_conc=zeros(time/dt,n);

ctc\_manure\_conc=zeros(time/dt,n);

%ctc concentration in LI and plasma and manure;

ctc\_st\_amt=zeros(time/dt,n);

ctc\_upsi\_amt=zeros(time/dt,n);

ctc\_rtsi\_amt=zeros(time/dt,n);

ctc\_tis\_amt=zeros(time/dt,n);

%ctc absolute amount;

%Ex (PD effect) storage arrays;

Er\_storage=zeros(time/dt,n);

Ei\_storage=zeros(time/dt,n);

Es\_storage=zeros(time/dt,n);

res\_eco\_growth=zeros(time/dt,n);

res\_eco\_outflow=zeros(time/dt,n);

res\_eco\_inflow=zeros(time/dt,n);

res\_eco\_cat\_pop=zeros(time/dt,n);

%see tables for parameter definitions;

%E coli population inter-individual stochasticity;

if POPSTOCHA==0

 Nmax=repmat(10^5.5,1,n);

 pop=Nmax/2;

 pr=repmat(0.35,1,n);

 pi=repmat(0.15,1,n);

 ps=repmat(0.50,1,n);

 r=repmat(0.17,1,n);

 bsi=repmat(0.002,1,n);

 bsr=repmat(0.001569317647,1,n);

 bir=repmat(0.0001452,1,n);

 lamda\_in=repmat(0.005,1,n);

 lamda\_out=repmat(0.01,1,n);

 alpha=repmat(0.02,1,n);

 start\_r=repmat(0.378,1,n);

 start\_i=repmat(0.164,1,n);

 start\_s=repmat(0.458,1,n);

else

 Nmax\_dist=makedist('Weibull',14.03,20.32);

 Nmax\_trunc=truncate(Nmax\_dist,3+7.59,8+7.59); %truncate (3,8)

 Nmax\_exp=random(Nmax\_trunc,1,n)-7.59; %shift by 7.59

 Nmax=10.^Nmax\_exp;

 % Nmax=10.^(wblrnd(14.03,20.32,1,n)-7.59);

 pop=Nmax\*unifrnd(.1,.9);

 pr=unifrnd(0.16,0.61,1,n);

 pi=unifrnd(0.02,0.15,1,n);

 ps=1-pr-pi;

 r=unifrnd(0.05,0.5,1,n);

 bsi=10.^(gamrnd(94.17,0.16,1,n)-22.57);

 bsr=10.^(gamrnd(94.17,0.16,1,n)-22.57);

 bir=10.^(gamrnd(94.17,0.16,1,n)-22.57);

 lamda\_in=unifrnd(0.001,0.01,1,n);

 lamda\_out=unifrnd(0.01,0.02,1,n);

 alpha=unifrnd(0,0.03,1,n);

 start\_r=unifrnd(0.16,0.61,1,n);

 start\_i=unifrnd(0.02,0.15,1,n);

 start\_s=1-start\_r-start\_i;

end

%PD inter-individual stochasticity;

if PDSTOCHA==0

 Emax=1;

 Hr=repmat(7.6,1,n);

 Hi=repmat(7.6,1,n);

 Hs=repmat(2.2,1,n);

 %EC50r=repmat(47.2,1,n);

 %EC50i=repmat(4.3,1,n);

 %EC50s=repmat(0.2,1,n);

 %if we use the anaerobic penalty to MIC:

 anaerobe\_s=repmat(-0.9,1,n);

 anaerobe\_i=repmat(-0.9,1,n);

 anaerobe\_r=repmat(-0.9,1,n);

 MICr=repmat(48,1,n)+anaerobe\_r;

 MICi=repmat(8,1,n)+anaerobe\_i;

 MICs=repmat(4,1,n)+anaerobe\_s;

 EC50r=2.^(-1.24+1.09\*log2(MICr));

 EC50i=2.^(-1.24+1.09\*log2(MICi));

 EC50s=2.^(-1.24+1.09\*log2(MICs));

else

 Emax=1;

 Hr=unifrnd(6.42,10,1,n);

 Hi=unifrnd(5.71,9.53,1,n);

 Hs=unifrnd(1.62,2.23,1,n);

 %EC50r=repmat(47.2,1,n);

 %EC50i=repmat(4.3,1,n);

 %EC50s=repmat(0.2,1,n);

 %if we use the anaerobic penalty to MIC:

 %anaerobe\_s=unifrnd(-1.3,0,1,n);

 %anaerobe\_i=unifrnd(-1.3,0,1,n);

 %anaerobe\_r=unifrnd(-1.3,0,1,n);

 %MICr=unifrnd(16,64,1,n)+anaerobe\_r;

 %MICi=unifrnd(4,16,1,n)+anaerobe\_i;

 %MICs=abs(unifrnd(0,4,1,n)+anaerobe\_s);

 MICr=unifrnd(16-1.3,128,1,n);

 MICi=unifrnd(4-1.3,16,1,n);

 MICs=unifrnd(0,4,1,n);

 EC50r=2.^(-1.24+1.09\*log2(MICr));

 EC50i=2.^(-1.24+1.09\*log2(MICi));

 EC50s=2.^(-1.24+1.09\*log2(MICs));

end

if PKSTOCHA==0

 delta=repmat(0.0333,1,n);

 gamma\_s=repmat(0.0715,1,n);

 gamma\_uppersi=repmat(0.333,1,n);

 gamma\_restsi=repmat(0.133,1,n);

 gamma\_li=repmat(0.133,1,n);

 %eta\_si=repmat(0.79,1,n);

 eta\_li=repmat(0.79,1,n);

 ka=repmat(0.0478,1,n);

 kpt=repmat(0.75,1,n);

 ktp=repmat(0.162,1,n);

 ke=repmat(1.14,1,n);

 Eb=repmat(0.515,1,n);

 Eu=1-Eb;

 Vp=repmat(57\*bw/1000,1,n);

 Vli=repmat(11,1,n);

else

 delta=betarnd(0.54,37.4,1,n);

 gamma\_s=unifrnd(0.0535,0.0895,1,n);

 gamma\_uppersi=unifrnd(0.25,0.416,1,n);

 gamma\_restsi=unifrnd(.1,.166,1,n);

 gamma\_li=unifrnd(0.1,0.166,1,n);

 %eta\_si=unifrnd(0.69,0.89,1,n);

 eta\_li=unifrnd(0.69,0.89,1,n);

 ka=repmat(0.0478,1,n);

 kpt=repmat(0.75,1,n);

 ktp=repmat(0.162,1,n);

 ke=repmat(1.14,1,n);

 Eb=unifrnd(0.39,0.64,1,n);

 Eu=1-Eb;

 Vp=repmat(57\*bw/1000,1,n);

 Vli=unifrnd(6,22,1,n);

end

for j=1:n

%PK compartments that are overwritten each simulation;

CTC=zeros(time/dt,6); %1-s,2-upper\_si,3-rest\_si,4-pl,5-tis,6-li

T=zeros(time/dt,1);

CTC\_feed=zeros(time/dt,1);

manure\_ctc\_conc=zeros(time/dt,1);

%E coli compartments that are overwritten each simulation;

pop\_ecoli=zeros(time/dt,1);

susc\_ecoli=zeros(time/dt,1);

resist\_ecoli=zeros(time/dt,1);

int\_ecoli=zeros(time/dt,1);

faeces=zeros(time/dt,1);

decay=zeros(time/dt,2);

%Initialization;

CTC(1,1)=0;

CTC(1,2)=0;

CTC(1,3)=0;

CTC(1,4)=0;

CTC(1,5)=0;

CTC(1,6)=0;

CTC\_feed(1,1)=0;

manure\_ctc\_conc(1,1)=0;

susc\_ecoli(1,1)=start\_s(1,j)\*pop(1,j);

resist\_ecoli(1,1)=start\_r(1,j)\*pop(1,j);

int\_ecoli(1,1)=start\_i(1,j)\*pop(1,j);

%starting amounts of SIR e coli;

pop\_ecoli(1,1)=susc\_ecoli(1,1)+resist\_ecoli(1,1)+int\_ecoli(1,1);

%starting values of stochastic arrays;

res\_eco\_li(1,j)=resist\_ecoli(1,1);

int\_eco\_li(1,j)=int\_ecoli(1,1);

susc\_eco\_li(1,j)=susc\_ecoli(1,1);

prop\_res\_eco(1,j)=resist\_ecoli(1,1)/pop\_ecoli(1,1);

prop\_int\_eco(1,j)=int\_ecoli(1,1)/pop\_ecoli(1,1);

prop\_susc\_eco(1,j)=susc\_ecoli(1,1)/pop\_ecoli(1,1);

total\_eco(1,j)=pop\_ecoli(1,1);

ctc\_li\_conc(1,j)=CTC(1,6)/Vli(1,j);

ctc\_pl\_conc(1,j)=CTC(1,4)/Vp(1,j);

ctc\_st\_amt(1,j)=CTC(1,1);

ctc\_upsi\_amt(1,j)=CTC(1,2);

ctc\_rtsi\_amt(1,j)=CTC(1,3);

ctc\_tis\_amt(1,j)=CTC(1,5);

ctc\_manure\_conc(1,j)=manure\_ctc\_conc(1,1);

i=2;

while i<(time/dt)+1

 T(i,1)=T(i-1,1)+dt;

 % T (time) variable is in increments of 0.1, units of hours;

 if T(i,1)>am\_start %start CTC

 if (mod(T(i,1),24) < 12)

 if METHODE==0

 CTCf=0;

 else

 if METHODE==1

 if (T(i,1)<(adc\_time+am\_start))

 CTCf=70/12; %ARLA dosage

 else

 CTCf=0;

 end

 elseif METHODE==2

 if (T(i,1)<(adc\_time+am\_start))

 CTCf=350/12; %ADC dosage

 else

 CTCf=0;

 end

 else

 if (T(i,1)<(adt\_time+am\_start))

 CTCf=22\*bw/12; %ADT dosage

 else

 CTCf=0;

 end

 end

 end

 else

 CTCf=0;

 end

 else

 CTCf=0;

 end

 CTC\_feed(i,1)=CTCf;

%intra-individual random components;

%st-stomach; in-income; po-degradation; up-upper SI; pl-plasma; re-rest SI; li-large intestine; ti-tissue; ou-outcome;

 if INTRASTOCHA==0

 rand\_u=0;

 rand\_b=0;

 rand\_stin=0;

 rand\_stpo=0;

 rand\_stup=0;

 rand\_uppo=0;

 rand\_uppl=0;

 rand\_upre=0;

 rand\_repo=0;

 rand\_reli=0;

 rand\_plpo=0;

 rand\_plti=0;

 rand\_tipl=0;

 rand\_tipo=0;

 rand\_liou=0;

 rand\_lipo=0;

 else

 rand\_u=normrnd(0,1);

 rand\_b=normrnd(0,1);

 rand\_stin=normrnd(0,1);

 rand\_stpo=normrnd(0,1);

 rand\_stup=normrnd(0,1);

 rand\_uppo=normrnd(0,1);

 rand\_uppl=normrnd(0,1);

 rand\_upre=normrnd(0,1);

 rand\_repo=normrnd(0,1);

 rand\_reli=normrnd(0,1);

 rand\_plpo=normrnd(0,1);

 rand\_plti=normrnd(0,1);

 rand\_tipl=normrnd(0,1);

 rand\_tipo=normrnd(0,1);

 rand\_liou=normrnd(0,1);

 rand\_lipo=normrnd(0,1);

 end

%bile and urine CTC;

B=ke(1,j)\*Eb(1,j)\*CTC(i-1,4)+(ke(1,j)\*Eb(1,j)\*CTC(i-1,4)/dt)^(1/2)\*rand\_b;

U=ke(1,j)\*Eu(1,j)\*CTC(i-1,4)+(ke(1,j)\*Eu(1,j)\*CTC(i-1,4)/dt)^(1/2)\*rand\_u;

%CTC flows in and out of each intestinal compartment;

 stomach\_income=CTCf+(CTCf/dt)^(1/2)\*rand\_stin;

 stomach\_deg=-(delta(1,j)\*CTC(i-1,1)+(delta(1,j)\*CTC(i-1,1)/dt)^(1/2)\*rand\_stpo);

 stom\_uppersi=gamma\_s(1,j)\*CTC(i-1,1)+(gamma\_s(1,j)\*CTC(i-1,1)/dt)^(1/2)\*rand\_stup;

 uppersi\_deg=-(delta(1,j)\*CTC(i-1,2)+(delta(1,j)\*CTC(i-1,2)/dt)^(1/2)\*rand\_uppo);

 uppersi\_plasma=ka(1,j)\*CTC(i-1,2)+(ka(1,j)\*CTC(i-1,2)/dt)^(1/2)\*rand\_uppl;

 uppersi\_restsi=gamma\_uppersi(1,j)\*CTC(i-1,2)+(gamma\_uppersi(1,j)\*CTC(i-1,2)/dt)^(1/2)\*rand\_upre;

 restsi\_deg=-(delta(1,j)\*CTC(i-1,3)+(delta(1,j)\*CTC(i-1,3)/dt)^(1/2)\*rand\_repo);

 restsi\_li=gamma\_restsi(1,j)\*CTC(i-1,3)+(gamma\_restsi(1,j)\*CTC(i-1,3)/dt)^(1/2)\*rand\_reli;

plasma\_deg=-(delta(1,j)\*CTC(i-1,4)+(delta(1,j)\*CTC(i-1,4)/dt)^(1/2)\*rand\_plpo);

plasma\_tissue=kpt(1,j)\*CTC(i-1,4)+(kpt(1,j)\*CTC(i-1,4)/dt)^(1/2)\*rand\_plti;

 tissue\_plasma=ktp(1,j)\*CTC(i-1,5)+(ktp(1,j)\*CTC(i-1,5)/dt)^(1/2)\*rand\_tipl;

 tissue\_deg=-(delta(1,j)\*CTC(i-1,5)+(delta(1,j)\*CTC(i-1,5)/dt)^(1/2)\*rand\_tipo);

li\_outcome=gamma\_li(1,j)\*CTC(i-1,6)+(gamma\_li(1,j)\*CTC(i-1,6)/dt)^(1/2)\*rand\_liou;

 li\_deg=-(delta(1,j)\*CTC(i-1,6)+(delta(1,j)\*CTC(i-1,6)/dt)^(1/2)\*rand\_lipo);

 %Equations for CTC in each compartment

if CTC(i-1,1)+dt\*(stomach\_income+stomach\_deg-stom\_uppersi)>0

 CTC(i,1)=CTC(i-1,1)+dt\*(stomach\_income+stomach\_deg-stom\_uppersi);

else

 CTC(i,1)=0;

end

if CTC(i-1,2)+dt\*(stom\_uppersi+B+uppersi\_deg-uppersi\_plasma-uppersi\_restsi)>0

 CTC(i,2)=CTC(i-1,2)+dt\*(stom\_uppersi+B+uppersi\_deg-uppersi\_plasma-uppersi\_restsi);

else

 CTC(i,2)=0;

end

if CTC(i-1,3)+dt\*(restsi\_deg+uppersi\_restsi-restsi\_li)>0

 CTC(i,3)=CTC(i-1,3)+dt\*(restsi\_deg+uppersi\_restsi-restsi\_li);

else

 CTC(i,3)=0;

end

if CTC(i-1,4)+dt\*(uppersi\_plasma-B-U+plasma\_deg-plasma\_tissue+tissue\_plasma)>0

 CTC(i,4)=CTC(i-1,4)+dt\*(uppersi\_plasma-B-U+plasma\_deg-plasma\_tissue+tissue\_plasma);

else

 CTC(i,4)=0;

end

if CTC(i-1,5)+dt\*(plasma\_tissue+tissue\_deg-tissue\_plasma)>0

 CTC(i,5)=CTC(i-1,5)+dt\*(plasma\_tissue+tissue\_deg-tissue\_plasma);

else

 CTC(i,5)=0;

end

if CTC(i-1,6)+dt\*(restsi\_li+li\_deg-li\_outcome)>0

 CTC(i,6)=CTC(i-1,6)+dt\*(restsi\_li+li\_deg-li\_outcome);

else

 CTC(i,6)=0;

end

 %12-30-16 new manure concentration;

 manure\_ctc\_conc(i,1)=((li\_outcome+U)/((0.001667\*bw)+(0.00075\*bw)))\*(1-eta\_li(1,j));

%above: 0.001667 is the defecation volume per kg bw per hr;

%PD equations;

 Es=1-((Emax\*((CTC(i-1,6)/Vli(1,j))\*(1-eta\_li(1,j))).^Hs(1,j))/((EC50s(1,j)).^Hs(1,j)+((CTC(i-1,6)/Vli(1,j))\*(1-eta\_li(1,j))).^Hs(1,j)));

 Er=1-((Emax\*((CTC(i-1,6)/Vli(1,j))\*(1-eta\_li(1,j))).^Hr(1,j))/((EC50r(1,j)).^Hr(1,j)+((CTC(i-1,6)/Vli(1,j))\*(1-eta\_li(1,j))).^Hr(1,j)));

 Ei=1-((Emax\*((CTC(i-1,6)/Vli(1,j))\*(1-eta\_li(1,j))).^Hi(1,j))/((EC50i(1,j)).^Hi(1,j)+((CTC(i-1,6)/Vli(1,j))\*(1-eta\_li(1,j))).^Hi(1,j)));

 %E coli population

 %intra-individual random components

 %growth

 if INTRASTOCHA==0

 rand\_g\_c=0;

 rand\_s\_g\_c\_1=0;

 rand\_s\_g\_c\_2=0;

 %g\_c=growth coefficient;

 rand\_r\_g\_c\_1=0;

 rand\_r\_g\_c\_2=0;

 rand\_i\_g\_c\_1=0;

 rand\_i\_g\_c\_2=0;

 rand\_pl\_si=0;

 rand\_pl\_sr=0;

 rand\_pl\_ir=0;

 %pl=plasmid in the e coli population model;

 else

 rand\_g\_c=normrnd(0,1);

 rand\_s\_g\_c\_1=normrnd(0,1);

 rand\_s\_g\_c\_2=normrnd(0,1);

 rand\_r\_g\_c\_1=normrnd(0,1);

 rand\_r\_g\_c\_2=normrnd(0,1);

 rand\_i\_g\_c\_1=normrnd(0,1);

 rand\_i\_g\_c\_2=normrnd(0,1);

 rand\_pl\_si=normrnd(0,1);

 rand\_pl\_sr=normrnd(0,1);

 rand\_pl\_ir=normrnd(0,1);

 end

%new plasmid transfer equations;

if pop\_ecoli(i-1,1)>0

 plasmid\_transfert\_si=bsi(1,j)\*susc\_ecoli(i-1,1)\*int\_ecoli(i-1,1)/(pop\_ecoli(i-1,1))+abs((bsi(1,j)\*susc\_ecoli(i-1,1)\*int\_ecoli(i-1,1)/pop\_ecoli(i-1,1))/dt)^(1/2)\*rand\_pl\_si;

else

 plasmid\_transfert\_si=0;

end

if pop\_ecoli(i-1,1)>0

 plasmid\_transfert\_sr=bsr(1,j)\*susc\_ecoli(i-1,1)\*resist\_ecoli(i-1,1)/(pop\_ecoli(i-1,1))+abs((bsr(1,j)\*susc\_ecoli(i-1,1)\*resist\_ecoli(i-1,1)/pop\_ecoli(i-1,1))/dt)^(1/2)\*rand\_pl\_sr;

else

 plasmid\_transfert\_sr=0;

end

if pop\_ecoli(i-1,1)>0

 plasmid\_transfert\_ir=bir(1,j)\*int\_ecoli(i-1,1)\*resist\_ecoli(i-1,1)/( pop\_ecoli(i-1,1))+abs((bir(1,j)\*int\_ecoli(i-1,1)\*resist\_ecoli(i-1,1)/pop\_ecoli(i-1,1))\*dt)^(1/2)\*rand\_pl\_ir;

else

 plasmid\_transfert\_ir=0;

end

 %e coli pop growth in cattle large intestine

susc\_cat\_population=r(1,j)\*(1-pop\_ecoli(i-1,1)/(Nmax(1,j)))\*Es\*susc\_ecoli(i-1,1)+abs((r(1,j)\*(1-pop\_ecoli(i-1,1)/(Nmax(1,j)))\*abs(Es)\*susc\_ecoli(i-1,1))/dt)^(1/2)\*rand\_s\_g\_c\_1+ps(1,j)\*lamda\_in(1,j)\*pop\_ecoli(i-1,1)+abs((ps(1,j)\*lamda\_in(1,j)\*pop\_ecoli(i-1,1)/dt))^(1/2)\*rand\_g\_c-lamda\_out(1,j)\*susc\_ecoli(i-1,1)-abs((lamda\_out(1,j)\*susc\_ecoli(i-1,1)/dt))^(1/2)\*rand\_s\_g\_c\_2;

res\_cat\_population=r(1,j)\*(1-alpha(1,j))\*(1-pop\_ecoli(i-1,1)/(Nmax(1,j)))\*Er\*resist\_ecoli(i-1,1)+abs((r(1,j)\*(1-alpha(1,j))\*(1-pop\_ecoli(i-1,1)/(Nmax(1,j)))\*abs(Er)\*resist\_ecoli(i-1,1)/dt))^(1/2)\*rand\_r\_g\_c\_1+pr(1,j)\*lamda\_in(1,j)\*pop\_ecoli(i-1,1)+abs((pr(1,j)\*lamda\_in(1,j)\*pop\_ecoli(i-1,1)/dt))^(1/2)\*rand\_g\_c-lamda\_out(1,j)\*resist\_ecoli(i-1,1)-abs((lamda\_out(1,j)\*resist\_ecoli(i-1,1)/dt))^(1/2)\*rand\_r\_g\_c\_2;

int\_cat\_population=r(1,j)\*(1-alpha(1,j))\*(1-pop\_ecoli(i-1,1)/(Nmax(1,j)))\*Ei\*int\_ecoli(i-1,1)+abs((r(1,j)\*(1-alpha(1,j))\*(1-pop\_ecoli(i-1,1)/(Nmax(1,j)))\*abs(Ei)\*int\_ecoli(i-1,1)/dt))^(1/2)\*rand\_i\_g\_c\_1+pi(1,j)\*lamda\_in(1,j)\*pop\_ecoli(i-1,1)+abs((pi(1,j)\*lamda\_in(1,j)\*pop\_ecoli(i-1,1)/dt))^(1/2)\*rand\_g\_c-lamda\_out(1,j)\*int\_ecoli(i-1,1)-abs((lamda\_out(1,j)\*int\_ecoli(i-1,1)/dt))^(1/2)\*rand\_i\_g\_c\_2;

%record the growth, inflow and outflow of resistant bacteria. Used to check for errors and inappropriate model behavior;

res\_eco\_growth(i,1)=r(1,j)\*(1-alpha(1,j))\*(1-pop\_ecoli(i-1,1)/(Nmax(1,j)))\*Er\*resist\_ecoli(i-1,1);

res\_eco\_inflow(i,1)=pr(1,j)\*lamda\_in(1,j)\*pop\_ecoli(i-1,1);

res\_eco\_outflow(i,1)=-lamda\_out(1,j)\*resist\_ecoli(i-1,1);

res\_eco\_cat\_pop(i,1)=res\_cat\_population;

 %Equations for e coli population

susc\_ecoli(i,1)=susc\_ecoli(i-1,1)+dt\*(susc\_cat\_population-plasmid\_transfert\_sr-plasmid\_transfert\_si);

resist\_ecoli(i,1)=resist\_ecoli(i-1,1)+dt\*(res\_cat\_population+plasmid\_transfert\_sr+plasmid\_transfert\_ir);

int\_ecoli(i,1)=int\_ecoli(i-1,1)+dt\*(int\_cat\_population+plasmid\_transfert\_si-plasmid\_transfert\_ir);

pop\_ecoli(i,1)=susc\_ecoli(i,1)+resist\_ecoli(i,1)+int\_ecoli(i,1);

%store data in arrays;

res\_eco\_li(i,j)=resist\_ecoli(i,1);

int\_eco\_li(i,j)=int\_ecoli(i,1);

susc\_eco\_li(i,j)=susc\_ecoli(i,1);

prop\_res\_eco(i,j)=resist\_ecoli(i,1)/pop\_ecoli(i,1);

prop\_int\_eco(i,j)=int\_ecoli(i,1)/pop\_ecoli(i,1);

prop\_susc\_eco(i,j)=susc\_ecoli(i,1)/pop\_ecoli(i,1);

total\_eco(i,j)=pop\_ecoli(i,1);

ctc\_li\_conc(i,j)=(CTC(i,6)/Vli(1,j))\*(1-eta\_li(1,j));

ctc\_pl\_conc(i,j)=CTC(i,4)/Vp(1,j);

ctc\_manure\_conc(i,j)=manure\_ctc\_conc(i,1);

ctc\_st\_amt(i,j)=CTC(i,1);

ctc\_upsi\_amt(i,j)=CTC(i,2);

ctc\_rtsi\_amt(i,j)=CTC(i,3);

ctc\_tis\_amt(i,j)=CTC(i,5);

%Ex storage array;

Er\_storage(i,j)=Er;

Ei\_storage(i,j)=Ei;

Es\_storage(i,j)=Es;

 i=i+1;

end

%calculations of SIR proportions and amounts at different time points;

%amount of e coli at 48 hours.;

t480\_susc\_eco\_li(1,j)=susc\_eco\_li(am\_start\*10,j);

t480\_res\_eco\_li(1,j)=res\_eco\_li(am\_start\*10,j);

t480\_int\_eco\_li(1,j)=int\_eco\_li(am\_start\*10,j);

t480\_total\_eco(1,j)=total\_eco(am\_start\*10,j);

%SIR in last 6 hours before tx;

prop\_res\_ss\_preCTC(1,j)=mean(prop\_res\_eco((am\_start-6)\*10:am\_start\*10,j));

prop\_int\_ss\_preCTC(1,j)=mean(prop\_int\_eco((am\_start-6)\*10:am\_start\*10,j));

prop\_susc\_ss\_preCTC(1,j)=mean(prop\_susc\_eco((am\_start-6)\*10:am\_start\*10,j));

%to find the average during tx;

if METHODE==3 %ADT

 tx\_end\_time=am\_start+adt\_time;

 %for ADT, average between tx start (day 2) and day 10 (3 days after tx

 %ends), which is when resistance peaks and LI CTC conc drops to almost 0;

 prop\_res\_ss\_CTC(1,j)=mean(prop\_res\_eco(am\_start\*10:(tx\_end\_time+72)\*10,j));

 prop\_int\_ss\_CTC(1,j)=mean(prop\_int\_eco(am\_start\*10:(tx\_end\_time+72)\*10,j));

 prop\_susc\_ss\_CTC(1,j)=mean(prop\_susc\_eco(am\_start\*10:(tx\_end\_time+72)\*10,j));

 %max during tx (see time frame above). added 5d to end of tx just to be sure got the max for every simulation since in ADT max occurs after treatment ends;

 [prop\_res\_max, prop\_res\_time\_max]=max(prop\_res\_eco(am\_start\*10:(tx\_end\_time+120)\*10,:));

 [prop\_int\_max, prop\_int\_time\_max]=max(prop\_int\_eco(am\_start\*10:(tx\_end\_time+120)\*10,:));

 [prop\_susc\_max, prop\_susc\_time\_max]=max(prop\_susc\_eco(am\_start\*10:(tx\_end\_time+120)\*10,:));

 %to find the average number of e coli during tx;

 total\_eco\_ss\_CTC(1,j)=mean(total\_eco(am\_start\*10:(tx\_end\_time+72)\*10,j));

else %ARLA, ADC or no CTC

 tx\_end\_time=am\_start+adc\_time;

 %for ADC and ARLA, an average also over 7 days (as above) but at the end of tx.;

 prop\_res\_ss\_CTC(1,j)=mean(prop\_res\_eco((tx\_end\_time-(7\*24))\*10:(tx\_end\_time)\*10,j));

 prop\_int\_ss\_CTC(1,j)=mean(prop\_int\_eco((tx\_end\_time-(7\*24))\*10:(tx\_end\_time)\*10,j));

 prop\_susc\_ss\_CTC(1,j)=mean(prop\_susc\_eco((tx\_end\_time-(7\*24))\*10:(tx\_end\_time)\*10,j));

 %max during tx;

 [prop\_res\_max, prop\_res\_time\_max]=max(prop\_res\_eco(am\_start\*10:(tx\_end\_time+120)\*10,:));

 [prop\_int\_max, prop\_int\_time\_max]=max(prop\_int\_eco(am\_start\*10:(tx\_end\_time+120)\*10,:));

 [prop\_susc\_max, prop\_susc\_time\_max]=max(prop\_susc\_eco(am\_start\*10:(tx\_end\_time+120)\*10,:));

 %to find the average number of e coli during tx (last 7 days of tx);

 total\_eco\_ss\_CTC(1,j)=mean(total\_eco((tx\_end\_time-(7\*24))\*10:(tx\_end\_time)\*10,j));

end

%proportion SIR averaged over the last 24 hours;

prop\_susc\_ss\_postCTC(1,j)=mean(prop\_susc\_eco((time-24)\*10:time\*10,j));

prop\_int\_ss\_postCTC(1,j)=mean(prop\_int\_eco((time-24)\*10:time\*10,j));

prop\_res\_ss\_postCTC(1,j)=mean(prop\_res\_eco((time-24)\*10:time\*10,j));

% e coli before treatment

total\_eco\_ss\_preCTC(1,j)=mean(total\_eco((am\_start-6)\*10:am\_start\*10,j));

%e coli after tx;

total\_eco\_ss\_postCTC(1,j)=mean(total\_eco((time-24)\*10:time\*10,j));

end