% RainbowTroutModel.m is a program that solves the system of ODEs that

% model the HPOL axis in rainbow trout. This program can be executed with

% either Octave or MATLAB. The output of this program is a vector of

% time points (in hours) with a corresponding solution matrix where each

% column represents a different protein in the HPOL axis. The input is the

% number of successive reproductive cycles you want to predict. The GnRH

% function provided (GnRH.xlsx) will run up to three consecutive cycles

% beginning from the first reproductive cycle.

% Both Octave and MATLAB require the files GnRH.xlsx and

% Parameters.xlsx to run; both of which are included as supplementary

% files. Further explanation of those files and how to customize the model

% for a specific data set can be found in Appendix S5.

% Running the code in MATLAB requires the Statistics Toolbox and the Curve

% Fitting Toolbox. Running the code in Octave requires the following

% packages: io, odepkg, splines, and statistics. When running the code in

% Octave uncomment the line of code which loads the packages needed.

function [Time, Solution]=RainbowTroutModel(cycles)

 % Packages needed for Octave. Uncomment the following line when using

 % Octave.

 % pkg load io; pkg load odepkg; pkg load splines; pkg load statistics;

 % Gets the parameters

 p=getParameters();

 % Creates the function for GnRH

 GnRHData=xlsread('GnRH.xlsx'); GnRH=pchip(GnRHData(:,1),GnRHData(:,2));

 save GnRHInput.mat GnRH

 % End time for one reproductive cycle (approximately 3 weeks after

 % ovulation)

 endTime = 361; % Days

 for i=1:cycles

 if i==1

 % Initial conditions for the initial spawning cycle

 mFSH\_0 = 36;

 mLH\_0 = 0.01;

 FSHP\_0 = 4.2;

 LHPit\_0 = 0;

 LHP\_0 = 0;

 E2\_0 = 0.5;

 DHP\_0 = 0;

 OAvg\_0 = 0.42;

 mR\_0 = 0;

 R\_0 = 0;

 ER\_0 = 0;

 mVTG\_0 = 0;

 VTGL\_0 = 0;

 VTGP\_0 = 0;

 VTGN\_0 = 0;

 DHPGreater\_0 = 0;

 FOMGreater\_0 = 0;

 % Note: Transit compartment initial conditions are equal to the

 % hormone being delayed.

 ICs = [mFSH\_0 mLH\_0 FSHP\_0 LHPit\_0 LHP\_0 E2\_0 DHP\_0 OAvg\_0 ...

 mR\_0 R\_0 ER\_0 mVTG\_0 VTGL\_0 VTGP\_0 VTGN\_0 FSHP\_0 FSHP\_0 ...

 FSHP\_0 FSHP\_0 FSHP\_0 FSHP\_0 FSHP\_0 E2\_0 E2\_0 E2\_0 LHP\_0...

 DHPGreater\_0 FOMGreater\_0];

 else

 % Sets the initial protein levels of the new cycle to the

 % protein levels at the end of the previous cycle and resets

 % the average oocyte growth for a new batch of oocytes and

 % resets the conditions required for ovulation.

 ICs = Y(length(Y),:);

 ICs(8)=OAvg\_0;

 ICs(27)=DHPGreater\_0;

 ICs(27)=FOMGreater\_0;

 end

 % Solves the Differential Equations (time is measured in hours)

 BegT=(i-1)\*endTime\*24; EndT=i\*endTime\*24;

 [T,Y] = ode23s(@(t,y) getDifferentialEquations (t, y, p),...

 [BegT EndT],ICs);

 if i==1

 Time=T;

 Solution=Y;

 else

 Time=[Time;T];

 Solution=[Solution;Y];

 end

 end

end

%% System of Differential Equations

function differentialEquations = getDifferentialEquations(t,y,p)

 % Variable names and corresponding position in the solutions matrix

 mFSH = y(1); %FSH beta subunit mRNA (Pituitary)

 mLH = y(2); %LH beta subunit mRNA (Pituitary)

 FSHP = y(3); %Follicle Stimulating Hormone (Plasma)

 LHPit = y(4); %Luteinizing Hormone (Pituitary)

 LHP = y(5); %Luteinizing Hormone (Plasma)

 E2 = y(6); %Estradiol-17beta (Plasma)

 DHP = y(7); %17alpha,20beta-hihydroxy-4-pregnen-3-one (Plasma)

 OAvg = y(8); %Average oocyte follicle diameter

 mR = y(9); %Estrogen receptor mRNA (Liver)

 R = y(10); %Estrogen receptor (Liver)

 ER = y(11); %Estrogen receptor complex (Liver)

 mVTG = y(12); %Vitellogenin mRNA (Liver)

 VTGL = y(13); %Vitellogenin protein (Liver)

 VTGP = y(14); %Vitellogenin protein (Plasma)

 VTGN = y(15); %Vitellogenin protein (Other)

 TCFSHP1 = y(16); %Transit compartment FSH to E2 (1)

 TCFSHP2 = y(17); %Transit compartment FSH to E2 (2)

 TCFSHP3 = y(18); %Transit compartment FSH to E2 (3)

 TCFSHP4 = y(19); %Transit compartment FSH to E2 (4)

 TCFSHP5 = y(20); %Transit compartment FSH to E2 (5)

 TCFSHP6 = y(21); %Transit compartment FSH to E2 (6)

 TCFSHP7 = y(22); %Transit compartment FSH to E2 (7)

 TCE21 = y(23); %Transit compartment E2 to mLH (1)

 TCE22 = y(24); %Transit compartment E2 to mLH (2)

 TCE23 = y(25); %Transit compartment E2 to mLH (3)

 TCLHP = y(26); %Transit compartment LH to DHP (1)

 DHPGreater = y(27); %Total time DHP has been higher than DHP\_final

 FOMGreater = y(28); %Total time FOM has been higher than FOM\_final

 % Input value for GnRH - estimated from the experimental data

 GnRH = getGnRHInput(t);

 % Conditions for Ovulation. When DHP is greater than DHP\_final,

 % DHPbigger will be positive and when FOM is greater than FOM\_final

 % then FOMbigger will be positive. When both DHPGreater and FOMGreater

 % are positive Ovulation will equal 1, otherwise it is 0. The

 % approximation of the Heaviside is used to satisfy uniqueness and

 % existence of solutions.

 Ovulation=Hsidecont(DHPGreater\*FOMGreater);

 % Oocyte growth stages

 [k,l]=WeibullParameters(OAvg,p,Ovulation);

 S1=wblcdf(p.s1,k,l);

 S2=wblcdf(p.s2,k,l)-wblcdf(p.s1,k,l);

 S3=wblcdf(p.s3,k,l)-wblcdf(p.s2,k,l);

 S4=wblcdf(p.s4,k,l)-wblcdf(p.s3,k,l);

 S5=wblcdf(p.s5,k,l)-wblcdf(p.s4,k,l);

 S6=(wblcdf(p.s6,k,l)-wblcdf(p.s5,k,l))\*(1-Ovulation);

 SFOM=(1-wblcdf(p.s6,k,l))\*(1-Ovulation);

 % The sytem of ODEs

 differentialEquations = [...

 p.ks\_mFSH\*(1+p.alpha\_mFSH\_GnRH\*GnRH)-p.kd\_mFSH\*mFSH %1: mFSH

 p.ks\_mLH\*(1+p.alpha\_mLH\_GnRH\*GnRH+p.alpha\_mLH\_E2\*TCE23)-...

 p.kd\_mLH\*mLH %2: mLH

 (1/p.V\_FSH)\*(p.w\_Pit\*p.ks\_FSH\*mFSH-p.Cl\_FSH\*FSHP) %3: FSH\_P

 p.ks\_LH\*mLH-p.kd\_LH\*LHPit-p.kr\_LH\*(LHPit-p.N\_E2\*E2-p.N\_DHP\*DHP)\*...

 Hside(LHPit-p.N\_E2\*E2-p.N\_DHP\*DHP)\*...

 (E2^p.n\_E2/(E2^p.n\_E2+p.T\_E2\_LH^p.n\_E2)) %4: LH\_Pit

 (1/p.V\_LH)\*(p.w\_Pit\*p.kr\_LH\*(LHPit-p.N\_E2\*E2-p.N\_DHP\*DHP)\*...

 Hside(LHPit-p.N\_E2\*E2-p.N\_DHP\*DHP)\*(E2^p.n\_E2/(E2^p.n\_E2+...

 p.T\_E2\_LH^p.n\_E2))-p.Cl\_LH\*LHP) %5: LH\_P

 (1/p.V\_E2\*(p.n\_oocyte\*(p.k\_E2+TCFSHP7\*(p.Cl\_E2\_S2\*S2+...

 p.Cl\_E2\_S3\*S3+p.Cl\_E2\_S4\*S4+p.Cl\_E2\_S5\*S5+p.Cl\_E2\_S6\*S6))-...

 p.Cl\_E2\*E2)) %6: E2

 (1/p.V\_DHP)\*(p.n\_oocyte\*(TCLHP\*(p.Cl\_DHP\_S2\*S2+p.Cl\_DHP\_S3\*S3+...

 p.Cl\_DHP\_S4\*S4+p.Cl\_DHP\_S5\*S5+p.Cl\_DHP\_S6\*S6+...

 p.Cl\_DHP\_SFOM\*(SFOM)))-p.Cl\_DHP\*DHP) %7: DHP

 (p.k\_NV\_OAvg\*(S1+S2+SFOM)+p.k\_V\_OAvg\*p.Cl\_VTG\_Seq\*(1+FSHP/(FSHP+...

 p.T\_Seq\_FSH))\*(S3+S4+S5+S6)\*VTGP) %8: O\_Avg

 p.ks\_mR\*(1+ER\*p.alpha\_mR\_ER)-p.kd\_mR\*mR %9: mR

 p.ks\_R\*mR-p.kd\_R\*R-p.kon\_ER\*E2\*R+p.koff\_ER\*ER %10: R

 p.kon\_ER\*E2\*R-(p.koff\_ER+p.kd\_ER)\*ER %11: ER

 p.ks\_mVTG\*(1+p.alpha\_mVTG\_ER\*ER)-p.kd\_mVTG\*mVTG %12: mVTG

 p.ks\_VTG\*(mVTG/p.N\_mVTG)^(p.gamma)-p.kr\_VTG\*VTGL %13: VTG\_L

 (p.kr\_VTG\*VTGL\*p.w\_L+p.Cl\_VTG\_Trans\*VTGN-(p.Cl\_VTG\_Trans+...

 p.Cl\_VTG\_Seq\*((1+FSHP/(FSHP+p.T\_Seq\_FSH))\*(S3+S4+S5+S6))+...

 p.Cl\_VTG)\*VTGP )/p.V\_VTG\_P %14: VTG\_P

 (p.Cl\_VTG\_Trans\*VTGP-p.Cl\_VTG\_Trans\*VTGN)/p.V\_VTG\_N %15: VTG\_N

 (7/p.D\_FSH\_E2)\*(FSHP-TCFSHP1) %16: TCFSHP1

 (7/p.D\_FSH\_E2)\*(TCFSHP1-TCFSHP2) %17: TCFSHP2

 (7/p.D\_FSH\_E2)\*(TCFSHP2-TCFSHP3) %18: TCFSHP3

 (7/p.D\_FSH\_E2)\*(TCFSHP3-TCFSHP4) %19: TCFSHP4

 (7/p.D\_FSH\_E2)\*(TCFSHP4-TCFSHP5) %10: TCFSHP5

 (7/p.D\_FSH\_E2)\*(TCFSHP5-TCFSHP6) %21: TCFSHP6

 (7/p.D\_FSH\_E2)\*(TCFSHP6-TCFSHP7) %22: TCFSHP7

 (3/p.D\_E2\_mLH)\*(E2-TCE21) %23: TCE21

 (3/p.D\_E2\_mLH)\*(TCE21-TCE22) %24: TCE22

 (3/p.D\_E2\_mLH)\*(TCE22-TCE23) %25: TCE23

 (1/p.D\_LH\_DHP)\*(LHP-TCLHP) %26: TCLHP

 Hsidecont(DHP-p.DHP\_final) %27: DHPbigger

 Hsidecont(SFOM-p.FOM\_final) % 28: FOMbigger

 ];

 % Eliminates the small imaginary part sometimes caclulated as a result

 % of approximating the solution for the system of odes.

 differentialEquations=real(differentialEquations);

end

%% Heaviside function

function Hside = Hside(t)

 if t>0

 Hside = 1;

 else

 Hside = 0;

 end

end

%% Continuous approximation of the Heaviside function

% Uses the Logistic function and is adjusted to have an ouput of 0 when

% evaluated at t=0.

function Hsidecont = Hsidecont(t)

 Hsidecont = 1/(1+exp(-100000000\*(t-10^(-3))));

end

%% Parameters for the Weibull Distribution

% Uses Nelder-Mead simplex method optimization to solve for the Weibull

% distribution parameters. The mean is defined by O\_Avg and the variance

% is defined by the stages.

function [k l]=WeibullParameters(mean,p,Ovulation)

 options = optimset('MaxFunEvals',5000);

 pars = fminsearch(@(a) WeibullParameterConditions([a(1),a(2)],mean,...

 p,Ovulation),[mean,1],options);

 k=pars(1); l=pars(2);

end

function obj = WeibullParameterConditions(abvector,mean,p,Ovulation)

 k = abvector(1); l = abvector(2);

 [m v]=wblstat(k,l);

 if Ovulation<1

 S1=wblcdf(p.s1,k,l);

 S2=wblcdf(p.s2,k,l)-wblcdf(p.s1,k,l);

 S3=wblcdf(p.s3,k,l)-wblcdf(p.s2,k,l);

 S4=wblcdf(p.s4,k,l)-wblcdf(p.s3,k,l);

 S5=wblcdf(p.s5,k,l)-wblcdf(p.s4,k,l);

 S6=wblcdf(p.s6,k,l)-wblcdf(p.s5,k,l);

 SFOM=1-wblcdf(p.s6,k,l);

 else

 S1=0; S2=0; S3=0; S4=0; S5=0; S6=0; SFOM=0;

 end

 % Oocyte Variance

 var=(p.alpha\_OVar\_S1\*S1+p.alpha\_OVar\_S2\*S2+p.alpha\_OVar\_S3\*S3+...

 p.alpha\_OVar\_S4\*S4+p.alpha\_OVar\_S5\*S5+p.alpha\_OVar\_S6\*S6)+...

 p.alpha\_OVar\_SFOM\*SFOM+p.alpha\_OVar\_S1\*Ovulation;

 obj= norm([mean var]-[m v]);

end

%% Parameters for the model

function p = getParameters()

 p = struct;

 [ParameterValues,ParameterNames]=xlsread('Parameters.xlsx');

 for i=1:length(ParameterNames)

 p.(ParameterNames{i}) = ParameterValues(i);

 end

end

%% Calls the GnRH function

function GnRH = getGnRHInput(t)

 load GnRHInput.mat; GnRH=fnval(t,GnRH);

 GnRH=(GnRH+abs(GnRH))/2;

end