```
ClearAll["Global`*"]
SetDirectory[NotebookDirectory[]];
SetOptions[$FrontEndSession, NotebookAutoSave → True];
NotebookSave[];
AppendTo[$Path, FileNameJoin[{$HomeDirectory, "Dropbox", "EpidCRNmodels"}]];
Needs["EpidCRN`"];
Get["HopfE`"];
```

In[3068]:=

```
(*Function:bd1-for RNs with only one strain*)
bd1[RN_, rts_] := Module[{spe, al, be, gam, Rv, RHS, def, var, par, cp, cv, ct, mS,
    mSi, inf, mod, K, eig, R0A, cDFE, RDFE, eq0, var0, E0, cE0, EA, RHSEj, eqEj,
    varEj, E1, E1NonDFE, E1Rational, Jx, Jy, ngm, isRationalSolutionQ, isDFEQ},
   {spe, al, be, gam, Rv, RHS, def} = extMat[RN];
   var = ToExpression[spe];
   RHS = gam.rts;
   par = Par[RHS, var];
   Print["RHS=", RHS // MatrixForm, " has var ", var, " par", par];
   cp = Thread[par > 0];
   cv = Thread[var ≥ 0];
   ct = Join[cp, cv];
   mS = minSiph[spe, asoRea[RN]];
   (*Get infection species and NGM*)mSi = Map[Flatten[Position[spe, #] & /@ #] &, mS];
   inf = Union[Flatten[mSi]];
   Print["minimal siphon ", mS⟦1⟧, " and invasion species are at positions: ", inf];
   (*Compute DFE (E0)*)cDFE = Flatten[Thread[ToExpression[#] → 0] & /@ mS];
   RDFE = RHS /. cDFE;
   eq0 = Thread[RDFE == 0];
   var0 = Complement[var, var⟦inf⟧];
   E0 = Join[Solve[eq0, var0] // Flatten, Thread[var⟦inf⟧ → 0]];
   cE0 = Select[Flatten[E0], (#⟦2⟧ == 0) &];
   Print["DFE solution E0: ", E0];
   (*Compute reproduction number*)mod = {RHS, var, par};
   ngm = NGM[mod, inf];
   Jx = ngm⟦1⟧ // FullSimplify /. Subscript[EpidCRN`Private`k, n_] :→ Subscript[k, n];
   Jy = ngm⟦5⟧ // FullSimplify /. Subscript[EpidCRN`Private`k, n_] :→ Subscript[k, n];
   K = ngm⟦4⟧ // FullSimplify /. Subscript[EpidCRN`Private`k, n_] :→ Subscript[k, n];
   Print["NGM K= ", K // MatrixForm];
   (*Get eigenvalues for single strain*)eig = Eigenvalues[K];
   R0A = Select[eig, (# =!= 0) &];
   Print["Reproduction function R0A: ", R0A];
   (*Compute boundary equilibrium (EA) for single strain*)EA = {};
   (*For the single strain:no other strains to set to 0*)RHSEj = RHS;
   eqEj = Thread[RHSEj == 0];
   varEj = var;
   AppendTo[EA, {eqEj, varEj}];
   (*Solve for fps (full solution set)*)fps = Solve[EA⟦1⟧⟦1⟧, EA⟦1⟧⟦2⟧];
```

```
    (*Helper function to detect rational solutions*)
    isRationalSolutionQ[sol_] := FreeQ[sol, Sqrt | Power[_, Except[_Integer]] | Root];
    (*Helper function to check if a solution is the DFE*)
    isDFEQ[sol_] := Module[{infectionVars, vals},
      (*Get the infection variable names from positions inf*)infectionVars = var⟦inf⟧;
      (*Get their values and simplify with parameter constraints*)
      vals = Simplify[infectionVars /. sol, cp];
      (*Check if all infection variables are zero*)And @@ ((# === 0) & /@ vals)];
    (*Filter for non-DFE rational solutions-this becomes E1*)
    E1NonDFE = Select[fps, (! isDFEQ[#]) &];
    E1 = Select[E1NonDFE, isRationalSolutionQ];
    (*Return with E1 (may be empty)*)
    {RHS, var, par, cp, mSi, Jx, Jy, E0, ngm, R0A, EA, E1}];
(*SEIR-B Model with Pathogen Dynamics*)(*Reaction Network*)

RN = {0 → "S", "S" + "B" → "e" + "B",
    "e" → "i", "e" → "e" + "B", "S" → 0, "B" → 0 , "e" → 0, "i" → 0};
(*Rate vector*)
rts = {Λ, β S B , e γe, ξ e, (γs + μ) S, μB B, μ e, (γi + μ) i};
Print["reactions and transition rates: ", Transpose[{RN, rts}] // MatrixForm]
(*Verify the ODE system*)
{RHS, var, par, cp, mSi, Jx, Jy, E0, ngm, R0A, EA, E1} = bd1[RN, rts];
E1 = E1 // Flatten;
E1 // FullSimplify
```

reactions and transition rates: $\begin{pmatrix} 0 \to S & \Lambda \\ B + S \to B + e & B\,S\,\beta \\ e \to i & e\,\gamma e \\ e \to B + e & e\,\xi \\ S \to 0 & S\,(\gamma s + \mu) \\ B \to 0 & B\,\mu B \\ e \to 0 & e\,\mu \\ i \to 0 & i\,(\gamma i + \mu) \end{pmatrix}$

RHS= $\begin{pmatrix} -B\,S\,\beta + \Lambda - S\,(\gamma s + \mu) \\ -B\,\mu B + e\,\xi \\ B\,S\,\beta - e\,\gamma e - e\,\mu \\ e\,\gamma e - i\,(\gamma i + \mu) \end{pmatrix}$ has var $\{S, B, e, i\}$ par $\{\beta, \gamma e, \gamma i, \gamma s, \Lambda, \mu, \mu B, \xi\}$

minimal siphon {B, e} and invasion species are at positions: {2, 3}

DFE solution E0: $\left\{i \to 0, S \to \frac{\Lambda}{\gamma s + \mu}, B \to 0, e \to 0\right\}$

NGM K= $\begin{pmatrix} 0 & 0 \\ \frac{S\,\beta}{\mu B} & \frac{S\,\beta\,\xi}{\gamma e\,\mu B + \mu\,\mu B} \end{pmatrix}$

Reproduction function R0A: $\left\{\frac{S\,\beta\,\xi}{(\gamma e + \mu)\,\mu B}\right\}$

Out[3073]=

$$\left\{S \to \frac{(\gamma e + \mu)\ \mu B}{\beta\ \xi},\ B \to -\frac{\gamma s + \mu}{\beta} + \frac{\Lambda\ \xi}{(\gamma e + \mu)\ \mu B},\right.$$
$$\left. e \to \frac{\Lambda}{\gamma e + \mu} - \frac{(\gamma s + \mu)\ \mu B}{\beta\ \xi},\ i \to \frac{-\gamma e\ (\gamma e + \mu)\ (\gamma s + \mu)\ \mu B + \beta\ \gamma e\ \Lambda\ \xi}{\beta\ (\gamma e + \mu)\ (\gamma i + \mu)\ \xi}\right\}$$

In[3055]:=

```
re = reCL[Reduce[Join[cp, Thread[(var /. E1) > 0]]]] // FullSimplify
```

Out[3055]=

$$\beta > \frac{(\gamma E + \mu)\ (\gamma S + \mu)\ \mu B}{\Lambda\ \xi}$$

In[3054]:=