EpidCRN Package Documentation

Context File for AI Assistant

August 28, 2025

1 Package Overview

EpidCRN is a Mathematica package for epidemiological models, which uses Chemical Reaction Network Theory (CRNT) methods. It aims to analyze models with unique disease-free boundary fixed point (DFE), possibly multi-strain (ie. with other boundary fixed points besides the DFE), and with unique positive fixed point (endemic).

2 Package Structure

2.1 Modular Organization

The package is split into subpackages to reduce monolithic complexity:

- `EpidCRN.wl` - Main loader package
- `Core.wl` - Basic network analysis (`EpidCRN‘Core‘`)
- `CRNT.wl` - Chemical reaction network theory (`EpidCRN‘CRNT‘`)
- `Boundary.wl` - NGM and boundary analysis (`EpidCRN‘Boundary‘`)
- `Bifurcation.wl` - Hopf bifurcations and parameter scanning (`EpidCRN‘Bifurcation‘`)
- `Siphons.wl` - Siphon and persistence analysis (`EpidCRN‘Siphons‘`)
- `Utils.wl` - Utility functions (`EpidCRN‘Utils‘`)

2.2 Dependency Chain

`Core` → `CRNT` → `Boundary` → `Bifurcation` → `Siphons`

3 Key Functions by Category

3.1 Core Functions (EpidCRN‘Core‘)

- `extMat[reactions]` - Master function extracting {species, α, β, γ, R_v, RHS, deficiency}
- `compToAsso[side]` - Parses reaction side to association of species → coefficients
- `extSpe[reactions]` - Extracts species list
- `asoRea[RN]` - Converts to association format with "Substrates"/"Products" keys

3.2 CRNT Functions (EpidCRN‘CRNT‘)

- `getComE[RN]` - Extracts {complexes, edges} from reaction network
- `IaFHJ[vertices, edges]` - Incidence matrix analysis, returns {matrix, tableForm}
- `IkFHJ[vertices, edges, rates]` - Ik matrix (n_reactions × n_complexes)
- `lapK[RN, rates]` - Laplacian matrix computation
- `SpeComInc[complexes, species]` - Species-complex incidence matrix

3.3 Boundary Analysis (EpidCRN‘Boundary‘)

- `NGM[mod, inf]` - Next Generation Matrix analysis
- `bd1[RN, rts]` - Single strain boundary analysis (DFE + 1 endemic)
- `bd2[RN, rts]` - Two strain analysis (DFE + 2 boundary + 1 coexistence)
- `DFE[mod, inf]` - Disease-Free Equilibrium computation
- `mRts[RN, ks]` - Mass action rates with parameter names

3.4 Bifurcation Analysis (EpidCRN‘Bifurcation‘)

- `fpHopf[RHS, var, par, p0val]` - Fixed point finder with Hopf analysis, returns angle = ArcTan[Re/Im]*180/Pi
- `simpleOptHopf[RHS, var, par, coP, optInd, numTries]` - Simple random search optimization for Hopf bifurcations
- `optHopf[RHS, var, par, coP, optInd, timeLimit, method, accGoal, precGoal, maxIter]` - Sophisticated NMaximize-based Hopf optimization
- `cont[RHS, var, par, p0val, stepSize, plotInd, bifInd, analyticalPlot]` - Continuation analysis with overlay capability
- `hopfD[curve]` - Hopf bifurcation detection from continuation curve data
- `scanPar[RHS, var, par, p0val, plotInd, gridRes, plot, ...]` - Comprehensive parameter space scanning with equilibrium classification
- `pertIC[equilibrium, var, factor, minq, n]` - Generate perturbed initial conditions
- `TS[RHS, var, par, p0val, tmax]` - Time series simulation using NDSolve
- `intEq[RHS, var, par, p0val, att]` - Interactive equilibrium finding with perturbations

4 Critical Workflows

4.1 Laplacian Matrix Construction

For ”invasion CRN” (network projected on classes that become 0 at DFE):

```
{complexes, edges} = getComE[RN];
incidenceMatrix = IaFHJ[complexes, edges][[1]];
ikMatrix = IkFHJ[complexes, edges, rates];
laplacian = ikMatrix; (* or incidenceMatrix . ikMatrix *)
```

4.2 Boundary Analysis Pipeline

```
(* Single strain *)
result1 = bd1[RN, rates];
{RHS, var, par, cp, mSi, Jx, Jy, E0, ngm, R0A, EA, E1} = result1;

(* Two strain *)
result2 = bd2[RN, rates];
{RHS, var, par, cp, mSi, Jx, Jy, E0, ngm, R0A, EA, E1t, E2t} = result2;
```

4.3 Hopf Bifurcation Detection

For detecting Hopf bifurcations through parameter optimization:

```
(* Simple approach - random search *)
{bestAngle, bestValues, finalP0Val} =
  simpleOptHopf[RHS, var, par, coP, {3,4}, 50];

(* Sophisticated approach - NMaximize *)
{bestAngle, bestValues, finalP0Val} =
  optHopf[RHS, var, par, coP, {3,4}, 120, "NelderMead", 4, 4, 500];

(* Angle interpretation: negative = stable focus, positive = Hopf *)
```

4.4 Continuation Analysis

For tracking equilibria along parameter paths:

```
(* Basic continuation along parameter 4 *)
curve = cont[RHS, var, par, p0val, 0.01, {3,4}, 4, None];

(* With analytical plot overlay *)
curve = cont[RHS, var, par, p0val, 0.01, {3,4}, 4, analyticalPlot];

(* Detect Hopf points from continuation curve *)
hopfPoints = hopfD[curve];
```

4.5 Parameter Space Scanning

For comprehensive equilibrium classification:

```
(* Grid mode scanning *)
{plot, errors, results} = scanPar[RHS, var, par, p0val, {1,2},
  30, Automatic, 0.01, 1/20, 0.5, 0.5, R01, R02, R21, R12];

(* Range mode scanning *)
{plot, errors, results} = scanPar[RHS, var, par, p0val, {1,2},
  Automatic, basePlot, 0.01, 0.05, 1.0, 1.0, R01, R02, R21, R12];
```

5 Important Implementation Details

5.1 Function Compatibility Issues

- `IaFHJ` and `IkFHJ` use `Table[gg[vert[[i]], edg[[j]]], ...]` NOT `Outer[gg, vert, edg]` due to Part specification errors

- Functions expect exact expression matching using `===` operator
- Species can be strings ("S1"), expressions ("I1" + "S1"), or symbols depending on context

5.2 Data Format Standards

- Reactions: {{`leftSide, rightSide`}, `...`} or {`leftSide -> rightSide, ...`}
- Species: List of strings {`"S1", "I1", "I2"`}
- Rates: List parallel to reactions {`k1, k2, k3`}
- Returns: Most functions return lists, not associations (user preference)

6 Key Concepts

6.1 Invasion Species

Species that become zero at Disease-Free Equilibrium, typically infection classes. Found using `minSiph[species, asoRea[RN]]`.

6.2 Boundary Analysis Types

- `bd1`: Single strain - expects DFE and one endemic equilibrium
- `bd2`: Two strain - expects DFE, two boundary points, one coexistence equilibrium

6.3 Matrix Conventions

- α: reactant stoichiometric matrix (species $\times$ reactions)
- β: product stoichiometric matrix
- $\gamma = \beta - \alpha$: net stoichiometric matrix
- Laplacian: follows CRNT conventions, NOT standard graph theory (row sums = 0)

7 Removed/Deprecated Functions

- `NGMs` - Removed (noted as treating "denominators and exponents incorrectly")
- `bdAnalG`, `bdAnalG0` - Removed (unclear differences from bd1/bd2)
- Multiple near-duplicate functions consolidated into master functions

8 File Structure

```
EpidCRN/
    EpidCRN.wl       (main package loader)
    Core.wl          (EpidCRN`Core` context)
    CRNT.wl          (EpidCRN`CRNT` context)
    Boundary.wl      (EpidCRN`Boundary` context)
    Bifurcation.wl   (EpidCRN`Bifurcation` context)
    Siphons.wl       (EpidCRN`Siphons` context)
    Utils.wl         (EpidCRN`Utils` context)
```

Critical: File names must match subcontext names exactly for `Get[]` to work automatically.

9 Usage Patterns

9.1 Loading

```
(* Load full package *)
Get["EpidCRN`"];

(* Load individual subpackage for testing *)
Get["EpidCRN`Core`"];
```

9.2 Typical Analysis Sequence

```
(* 1. Extract network structure *)
{species, alpha, beta, gamma, Rv, RHS, deficiency} = extMat[RN];

(* 2. Boundary analysis *)
boundaryResults = bd2[RN, rates];

(* 3. Parameter space scanning *)
{plot, errors, results} = scanPar[RHS, variables, parameters, p0val,
  {1,2}, 30, Automatic];

(* 4. Hopf bifurcation search *)
{bestAngle, bestValues, finalP0Val} =
  optHopf[RHS, variables, parameters, coP, {3,4}, 120];

(* 5. Continuation analysis *)
curve = cont[RHS, variables, parameters, finalP0Val, 0.01, {3,4}, 4];
hopfPoints = hopfD[curve];

(* 6. Laplacian for invasion dynamics *)
{laplacian, complexes, edges} = lapK[RN, rates];

(* 7. NGM analysis *)
mod = {RHS, variables, parameters};
ngmResults = NGM[mod, infectionIndices];
```

10 Notes for Future Development

- User strongly prefers list returns over association returns
- Focus on functions that work symbolically and numerically
- Avoid vague descriptions like "complete analysis" - specify what differs between functions
- Package grew organically with substantial duplication - consolidation ongoing
- Core functions like `extMat` are dependency roots for most other functions
- `Bifurcation` subpackage focuses on dynamical systems analysis - works numerically with NDSolve
- Hopf detection uses angle criterion: negative = stable focus, positive = Hopf bifurcation
- Parameter scanning supports both grid mode (fixed resolution) and range mode (adaptive stepping)
- Continuation analysis can overlay results on analytical plots when bifurcation parameter is in plot indices