

LEARN

“A Lyapunov-enabled analysis of biochemical reaction networks”

M. Ali Al-Radhawi¹, David Angeli^{1,3}, and Eduardo D. Sontag^{1, 4}

¹Departments of Bioengineering and of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115.

²Department of Electrical & Electronic Engineering, Imperial College London, London SW7 2AZ, UK.

³Dipartimento di Ingegneria dell’Informazione, University of Florence, Florence, Italy.

We describe the prerequisites of **LEARN**, the basic subroutines offered in **LEARN** and few example runs. We assume that the reader is familiar with [1].

1 Prerequisites

LEARN runs on MATLAB with the optimization and symbolic math toolboxes. Also, it needs the **cvx** package. The latest version of **cvx** is available on the link <http://cvxr.com/cvx/download/>. After download, the user must run **cvx_setup**. After **cvx_setup** reporting that **cvx** is successfully installed, **LEARN** should run without issues.

2 List of Subroutines

The following subroutines are available. Note that all the subroutines below take Γ as an input which is the stoichiometry matrix of the network. If the network has an autocatalytic reaction then both matrices A, B need to be entered. (see the Methods section in [1])

2.1 Main subroutines

- **LEARNmain(Gamma)**: Prints a basic report on the network which should be sufficient for most users. Examples will follow. Another parallel function, **LEARNmainplus(Gamma)**, is available which runs a more exhaustive RLF search, but it can be slower.

2.2 Basic subroutines

- **d=IsConservative(Gamma)**: Checks if the network is conservative. If it is, then the subroutine returns a positive vector $d \in \mathbb{R}_+^n$ such that $d^T \Gamma = 0$. If the network is not conservative then d returns a scalar 0.
- **v=IsAS1(Gamma)**: Checks if the stoichiometry matrix has a positive vector in its kernel. If it does, then the subroutine returns a positive vector $v \in \mathbb{R}_+^n$ such that $\Gamma v = 0$. If the network is not conservative then d returns a scalar 0.

- `[flag, deadlock]=checkSiphons(Gamma)`: Checks if there are critical siphons and deadlocks. Each output can be either 0 or 1.
- `flag=checkMnetwork(Gamma)`: Checks if the network is an M -network. The output is either 0 or 1.

2.3 Necessary Conditions

- `checkSiphonCondition(Gamma)`: Checks if the network violates the critical siphon necessary condition [1, Theorem 8]. It prints a brief report.
- `flag=SignPatternCheck(Gamma)`: Checks if the network violates the sign pattern necessary condition [2, Theorem 9]. The output is either 0 or 1.
- `flag=checkPmatreix(Gamma)`: Checks if the network violates the P matrix necessary condition [2, Theorem 8]. The output is either 0 or 1.
- `flag=RobustNondegeneracy(Gamma)`: Checks if the network has a robustly non-degenerate Jacobian [1, Theorem 7]. This only applies to networks that pass the P matrix test. The output is either 0 or 1.

2.4 Construction of RLFs

- `C=ConstructGraphical(Gamma)`: Checks if the network admits the Max-Min RLF as given in [1, Theorem 4]. The output is C . If the method fails then C will be an empty matrix.
- `C=ConstructIterate(Gamma)`: Checks if the network admits an RLF as given in [1, Theorem 3]. The output is C . If the method fails then C will be an empty matrix.
- `[C, cvx]=ConstructLP(Gamma, H2, w, c)`: Checks if a non-autocatalytic network admits an RLF as given in [1, Theorem 2]. The last three inputs are optional. The output is C and the flag `cvx` to indicate that the RLF has been certified to be convex. The second input is H_2 which are optional rows to add to the partitioning matrix $H = \Gamma$. The default value for H_2 is an empty matrix. The third input is w and it is a flag to constrain the search to Sum-of-Currents RLFs. The default value is 1, but it is set to 0 in the `LEARNmainplus` subroutine. The fourth input is a flag to constrain the RLF to be convex. The default value is 0 which is the recommended value.
- `[C, cvx]=ConstructLPauto(A, B, H2, w, c)`: Checks if an *autocatalytic* network admits an RLF as given in [1, Theorem 2]. The remaining input structure is similar to the previous subroutine.
- `[C]=ConstructCoP(Gamma, H2)`: Checks if a non-autocatalytic network admits an RLF as given in [1, Theorem 5]. The last input is optional. The output is a tensor of PWQ RLF matrices. The second input is H_2 which are optional rows to add to the partitioning matrix $H = \Gamma$. The default value for H_2 is an empty matrix.

2.5 Checking a candidate RLF

- `flag=CheckRLF(Gamma, C)`: Checks if $\tilde{V} = \max_k c_k^T r$ is an RLF for a non-autocatalytic network with the stoichiometry matrix Γ .

3 Examples

All the examples are included in the folder `examples`.

3.1 The double processive PTM cycle

This is the form of the input to `LEARN` for the network depicted in [1, Fig. 9-b].

```
Gamma=[
-1      1      0      0      0      0      0      1;
-1      1      0      0      0      0      1      0;
1      -1     -1      0      0      0      0      0;
0      0      0     -1      1      0      1      0;
0      0      0     -1      1      0      0      1;
0      0      0      1     -1     -1      0      0;
0      0      1      0      0      0     -1      0;
0      0      0      0      0      1      0     -1];
LEARNmain(Gamma)
```

Note that the stoichiometry matrix Γ can be easily written from a list of reactions. The output of `LEARN` is as follows:

```
-----
Welcome to LEARN v1.01, Jan 2020
Developed by M. Ali Al-Radhawi malirdwi@{northeastern.edu,mit.edu,
gmail.com}

LEARN tries to construct a Robust Lyapunov Function for a given
reaction network.
-----
The network has 8 species and 8 reactions.
The stoichiometric space is 5-dimensional.
The network has a positive vector in the kernel of the
stoichiometry matrix, i.e. it has the potential for positive
steady states.
The network is conservative.
The network has no critical siphons. It is structurally persistent.
-----
LEARN will check some necessary conditions
Necessary Condition # 1 ....
The critical siphon necessary condition is satisfied.
Necessary Condition # 2 ....
The sign pattern necessary condition is satisfied.
Necessary Condition # 3 ....
The P matrix necessary condition is satisfied.
-----
LEARN will search for a PWL RLF
Method # 1: Graphical Method ..
This is an M-network. The graphical criteria will be checked
```

Success!! A PWL RLF has been found.

The following is always a Lyapunov function for any monotone kinetics: $V(x) = || C \cdot R(x) ||_{\infty}$,

where C is given as follows:

0	0	1	0	0	-1	0	0
0	0	1	0	0	0	-1	0
0	0	1	0	0	0	0	-1
-1	1	1	0	0	0	0	0
0	0	1	-1	1	0	0	0
0	0	0	0	0	1	-1	0
0	0	0	0	0	1	0	-1
-1	1	0	0	0	1	0	0
0	0	0	-1	1	1	0	0
0	0	0	0	0	0	1	-1
-1	1	0	0	0	0	1	0
0	0	0	-1	1	0	1	0
-1	1	0	0	0	0	0	1
0	0	0	-1	1	0	0	1
1	-1	0	-1	1	0	0	0

The robust non-degeneracy test is passed.

Since the network is conservative and with no critical siphons then the following holds:

There exists a unique positive globally asymptotically stable steady state in each stoichiometric class.

Method # 2: Iterative Method ..

Success!! A PWL RLF has been found.

The following is always a Lyapunov function for any monotone kinetics: $V(x) = || C \cdot R(x) ||_{\infty}$,

where C is given as follows:

-1	1	0	0	0	0	0	1
-1	1	0	0	0	0	1	0
1	-1	-1	0	0	0	0	0
0	0	0	-1	1	0	1	0
0	0	0	-1	1	0	0	1
0	0	0	1	-1	-1	0	0
0	0	1	0	0	0	-1	0
0	0	0	0	0	1	0	-1
0	0	0	0	0	0	-1	1
0	0	-1	0	0	0	0	1
-1	1	0	0	0	1	0	0
0	0	0	0	0	-1	1	0
0	0	1	-1	1	0	0	0
0	0	-1	0	0	1	0	0
-1	1	0	1	-1	0	0	0

The robust non-degeneracy test is passed.

Since the network is conservative and with no critical siphons then the following holds:

There exists a unique positive globally asymptotically stable steady state in each stoichiometric class.

Method # 3: Linear Programming Method ..

The partition matrix H is set to the default choice H=the stoichiometry matrix ..

This method for constructing a PWL RLF has failed.

THE END.

3.2 The double distributive PTM cycle

This is the output of LEARNmain for the network depicted in [1, Fig. 9-d].

Welcome to LEARN v1.01, Jan 2020

?Developed by M. Ali Al-Radhawi malirdwi@{northeastern.edu,mit.edu, gmail.com}

LEARN tries to construct a Robust Lyapunov Function for a given reaction network.

The network has 9 species and 12 reactions.

The stoichiometric space is 6-dimensional.

The network has a positive vector in the kernel of the stoichiometry matrix, i.e. it has the potential for positive steady states.

The network is conservative.

The network has no critical siphons. It is structurally persistent.

LEARN will check some necessary conditions

Necessary Condition # 1

The critical siphon necessary condition is satisfied.

Necessary Condition # 2

The sign pattern necessary condition is satisfied.

Necessary Condition # 3

The P matrix necessary condition is violated. A PWL RLF does not exist

LEARN will search for a PWL RLF

Method # 1: Graphical Method ..

This is not an M-network. Method # 1 is not applicable.

Method # 2: Iterative Method ..

This method for constructing a PWL RLF has failed.

```

Method # 3: Linear Programming Method ..
The partition matrix H is set to the default choice H=the
stoichiometry matrix ..
This method for constructing a PWL RLF has failed.
THE END.

```

3.3 The McKeithan Network

This is the output of LEARNmain for the network depicted in [1, Fig. 11-a] with $N = 2$.

```

-----
Welcome to LEARN v1.01, Jan 2020
Developed by M. Ali Al-Radhawi malirdwi@{northeastern.edu,mit.edu,
gmail.com}

LEARN tries to construct a Robust Lyapunov Function for a given
reaction network.
-----
The network has 5 species and 6 reactions.
The stoichiometric space is 3-dimensional.
The network has a positive vector in the kernel of the
stoichiometry matrix, i.e. it has the potential for positive
steady states.
The network is conservative.
The network has no critical siphons. It is structurally persistent.
-----
LEARN will check some necessary conditions
Necessary Condition # 1 ....
The critical siphon necessary condition is satisfied.
Necessary Condition # 2 ....
The sign pattern necessary condition is satisfied.
Necessary Condition # 3 ....
The P matrix necessary condition is satisfied.
-----
LEARN will search for a PWL RLF
Method # 1: Graphical Method ..
This is not an M-network. Method # 1 is not applicable.

-----
Method # 2: Iterative Method ..
Success!! A PWL RLF has been found.
The following is always a Lyapunov function for any monotone
kinetics:  $V(x) = || C * R(x) ||_{\infty}$ ,
where C is given as follows:
-1      1      0      0      1      1
-1      1      0      0      1      1
1      -1     -1      0      0      0
0      0      1     -1     -1      0

```

0	0	0	1	0	-1
0	0	-1	0	1	1
-1	1	1	-1	0	1
-1	1	0	1	1	0

The robust non-degeneracy test is passed.

Since the network is conservative and with no critical siphons then the following holds:

There exists a unique positive globally asymptotically stable steady state in each stoichiometric class.

Method # 3: Linear Programming Method ..

The partition matrix H is set to the default choice H=the stoichiometry matrix ..

Success!! A PWL RLF has been found.

The following is always a Lyapunov function for any monotone kinetics: $V(x) = || C \cdot R(x) ||_{\infty}$,

where C is given as follows:

0	0	0	1	0	-1
0	0	1	-1	-1	0
0	0	1	-0	-1	-1
1	-1	-1	0	0	0
1	-1	-1	1	0	-1
1	-1	-0	-1	-1	0
1	-1	-0	-0	-1	-1

The robust non-degeneracy test is passed.

Since the network is conservative and with no critical siphons then the following holds:

There exists a unique positive globally asymptotically stable steady state in each stoichiometric class.

Please note that this function is a Sum-of-Currents RLF

which can alternatively be written

as $V(x) = \sum_i x_i | \dot{x}_i |$, where $x_i = [x_{i_1} \dots x_{i_n}] =$

1	1	2	2	2
---	---	---	---	---

THE END.

References

1. M. Ali Al-Radhawi, David Angeli, and Eduardo Sontag. A computational framework for a Lyapunov-enabled analysis of biochemical reaction networks. *bioRxiv:696716*, 2019.
2. M. Ali Al-Radhawi and David Angeli. New approach to the stability of chemical reaction networks: Piecewise linear in rates lyapunov functions. *IEEE Transactions on Automatic Control*, 61(1):76–89, 2016.