

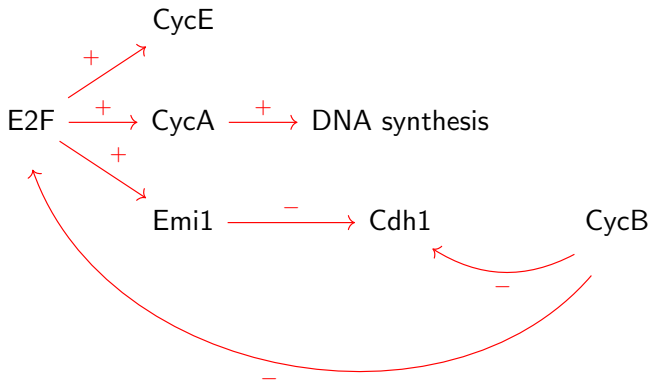
Emergent Feedback Loops from Crosstalk in Directed Labelled Graphs

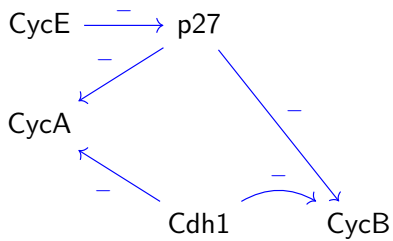
Aditty Chaudhuri (joint work with John C. Baez)

September 25, 2026

The general idea

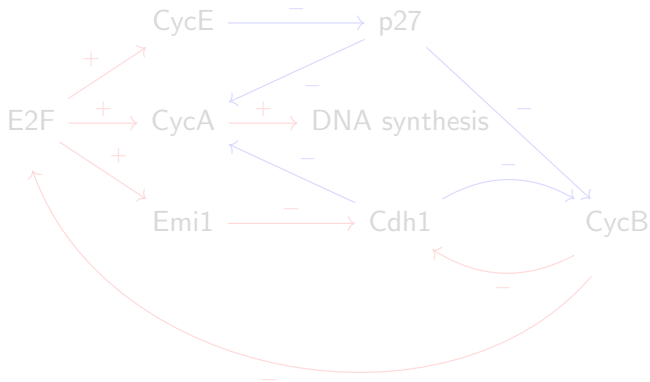
Let us consider the following pair of regulatory interactions within a cell, which are crucial for DNA synthesis.





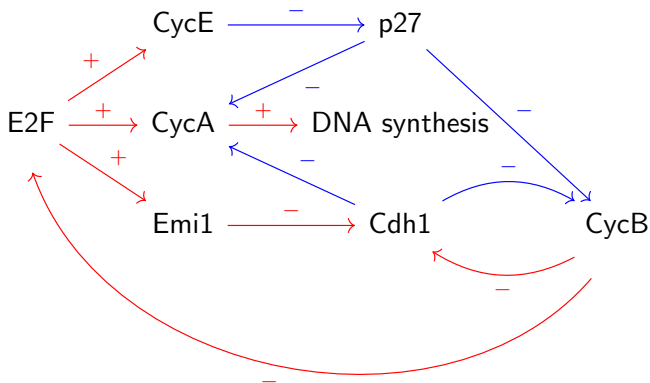
Observe that there are no feedback loops present in each of them.

However, when we compose them along the common nodes viz. proteins CycE, CycA, Cdh1 and CycB, we obtain feedback loops, which were absent before, as is evident from the diagram below.



Observe that there are no feedback loops present in each of them.

However, when we compose them along the common nodes viz. proteins CycE, CycA, Cdh1 and CycB, we obtain feedback loops, which were absent before, as is evident from the diagram below.



Note that in the above diagram, every directed loop is composed of mixed colors (red and blue), and there is not a single monochromatic directed cycle present.

We call these new feedback loops **emergent feedback loops**, which we *expect* to give rise to *new emergent behaviour* in the combined system of networks.

Note that in the above diagram, every directed loop is composed of mixed colors (red and blue), and there is not a single monochromatic directed cycle present.

We call these new feedback loops **emergent feedback loops**, which we *expect* to give rise to *new emergent behaviour* in the combined system of networks.

Biological motivation for studying emergent feedback loops

Biological pathway crosstalk is a natural source of emergent feedback because cellular pathways rarely act in isolation: when distinct modules share components or regulate one another, new feedback loops can appear in the combined network even if neither module contains a loop on its own. These emergent loops can drive important biological behaviors such as sustained signaling, growth control, and developmental patterning.

A **compelling evidence of this phenomenon** is provided by the *Wnt-ERK crosstalk*, where a hidden oncogenic positive feedback loop embedded in the combined network generates bistability in both pathways and can sustain activation even after the original input signal has been withdrawn, a dynamical behavior with direct implications for oncogenesis.¹

¹Kim, D., Rath, O., Kolch, W. et al. A hidden oncogenic positive feedback loop caused by crosstalk between Wnt and ERK Pathways. *Oncogene* 26, 4571-4579 (2007). <https://doi.org/10.1038/sj.onc.1210230>

Biological motivation for studying emergent feedback loops

Biological pathway crosstalk is a natural source of emergent feedback because cellular pathways rarely act in isolation: when distinct modules share components or regulate one another, new feedback loops can appear in the combined network even if neither module contains a loop on its own. These emergent loops can drive important biological behaviors such as sustained signaling, growth control, and developmental patterning.

A compelling evidence of this phenomenon is provided by the *Wnt-ERK crosstalk*, where a hidden oncogenic positive feedback loop embedded in the combined network generates bistability in both pathways and can sustain activation even after the original input signal has been withdrawn, a dynamical behavior with direct implications for oncogenesis.¹

¹Kim, D., Rath, O., Kolch, W. et al. A hidden oncogenic positive feedback loop caused by crosstalk between Wnt and ERK Pathways. *Oncogene* 26, 4571?4579 (2007). <https://doi.org/10.1038/sj.onc.1210230>

Question we ask

How to characterise the notion of emergent feedback loops structurally?

We approached the question *algebraically* in a general setting of **monoid labeled graphs**, that realise standard regulatory networks in special cases.

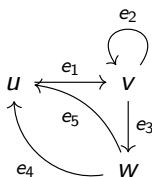
Question we ask

How to characterise the notion of emergent feedback loops structurally?

We approached the question *algebraically* in a general setting of **monoid labeled graphs**, that realise standard regulatory networks in special cases.

Graphs

For us, **graphs** are tuples $G = (s, t: E \rightarrow V)$ which consists of a set of **vertices** V and a set of **edges** E together with a **source map** $s: E \rightarrow V$ and a **target map** $t: E \rightarrow V$.



We define a **map of graphs** $f = (f_0, f_1)$ from a graph $G = (E, V, s, t)$ to a graph $G' = (E', V', s', t')$ to be a pair of functions $f_0: V \rightarrow V'$ and $f_1: E \rightarrow E'$ such that the following two diagrams commute:

$$\begin{array}{ccc} E & \xrightarrow{f_1} & E' \\ s \downarrow & & \downarrow s' \\ V & \xrightarrow{f_0} & V' \end{array} \qquad \begin{array}{ccc} E & \xrightarrow{f_1} & E' \\ t \downarrow & & \downarrow t' \\ V & \xrightarrow{f_0} & V' \end{array}.$$

The collection of graphs and their morphisms form a category which we denote as $\mathbf{Gph}$.

Monoid labeled graphs

Although a graph captures the interconnection between a set of elements, it does not say anything about the *types of interconnection*.

This leads us to the framework of **monoid labeled graphs**.

What is a monoid? A **monoid** is a set M together with a binary operation

$$\cdot : M \times M \rightarrow M,$$

which enjoys associativity and the existence of identity element 1 with the associated unitality law.

Given a monoid $(M, \cdot, 1)$, we define a **M -labeled graph** (G, ℓ) to be a graph $G = (E, V, s, t)$ equipped with a **labeling map** $\ell : E \rightarrow M$. We call $(M, \times, 1)$ to be the monoid of **labels**.

Monoid labeled graphs

Although a graph captures the interconnection between a set of elements, it does not say anything about the *types of interconnection*.

This leads us to the framework of **monoid labeled graphs**.

What is a monoid? A **monoid** is a set M together with a binary operation

$$\cdot : M \times M \rightarrow M,$$

which enjoys associativity and the existence of identity element 1 with the associated unitality law.

Given a monoid $(M, \cdot, 1)$, we define a **M -labeled graph** (G, ℓ) to be a graph $G = (E, V, s, t)$ equipped with a **labeling map** $\ell : E \rightarrow M$. We call $(M, \cdot, 1)$ to be the monoid of **labels**.

Monoid labeled graphs

Although a graph captures the interconnection between a set of elements, it does not say anything about the *types of interconnection*.

This leads us to the framework of **monoid labeled graphs**.

What is a monoid? A **monoid** is a set M together with a binary operation

$$\cdot : M \times M \rightarrow M,$$

which enjoys associativity and the existence of identity element 1 with the associated unitality law.

Given a monoid $(M, \cdot, 1)$, we define a **M -labeled graph** (G, ℓ) to be a graph $G = (E, V, s, t)$ equipped with a **labeling map** $\ell : E \rightarrow M$. We call $(M, \times, 1)$ to be the monoid of **labels**.

Monoid graded categories

- A labeled edge $u \xrightarrow{\ell(e)} v$ models a **direct effect of type $\ell(e)$** of u on v .

Why 'monoid' labeled graphs ?

It lets us model **indirect typed effects of an entity on another**, via the monoid graded category associated to a monoid labeled graph.

In particular, **the M -graded category associated to a M -labeled graph (G, ℓ)** is $\text{Free}_M(G, \ell)$ which has $\text{Free}(G)$, the *path category of the graph G* as its underlying category, with the labeling of any morphism

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_n$$

in $\text{Free}(G)$ being the product $\ell(e_n) \cdots \ell(e_1) \in M$.

- We like to interpret that the vertex v_0 has an indirect effect of type $\ell(e_n) \cdots \ell(e_1)$ on the vertex v_n .

Monoid graded categories

- A labeled edge $u \xrightarrow{\ell(e)} v$ models a **direct effect of type $\ell(e)$** of u on v .

Why 'monoid' labeled graphs ?

It lets us model **indirect typed effects of an entity on another**, via the monoid graded category associated to a monoid labeled graph.

In particular, **the M -graded category associated to a M -labeled graph (G, ℓ)** is $\text{Free}_M(G, \ell)$ which has $\text{Free}(G)$, the *path category of the graph G* as its underlying category, with the labeling of any morphism

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_n$$

in $\text{Free}(G)$ being the product $\ell(e_n) \cdots \ell(e_1) \in M$.

- We like to interpret that the vertex v_0 has an indirect effect of type $\ell(e_n) \cdots \ell(e_1)$ on the vertex v_n .

Monoid graded categories

- A labeled edge $u \xrightarrow{\ell(e)} v$ models a **direct effect of type $\ell(e)$** of u on v .

Why 'monoid' labeled graphs ?

It lets us model **indirect typed effects of an entity on another**, via the monoid graded category associated to a monoid labeled graph.

In particular, **the M -graded category associated to a M -labeled graph (G, ℓ)** is $\text{Free}_M(G, \ell)$ which has $\text{Free}(G)$, the *path category of the graph G* as its underlying category, with the labeling of any morphism

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_n$$

in $\text{Free}(G)$ being the product $\ell(e_n) \cdots \ell(e_1) \in M$.

- We like to interpret that the vertex v_0 has an indirect effect of type $\ell(e_n) \cdots \ell(e_1)$ on the vertex v_n .

Monoid graded categories

- A labeled edge $u \xrightarrow{\ell(e)} v$ models a **direct effect of type $\ell(e)$** of u on v .

Why 'monoid' labeled graphs ?

It lets us model **indirect typed effects of an entity on another**, via the monoid graded category associated to a monoid labeled graph.

In particular, **the M -graded category associated to a M -labeled graph (G, ℓ)** is $\text{Free}_M(G, \ell)$ which has $\text{Free}(G)$, the *path category of the graph G* as its underlying category, with the labeling of any morphism

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_n$$

in $\text{Free}(G)$ being the product $\ell(e_n) \cdots \ell(e_1) \in M$.

- We like to interpret that the vertex v_0 has an indirect effect of type $\ell(e_n) \cdots \ell(e_1)$ on the vertex v_n .

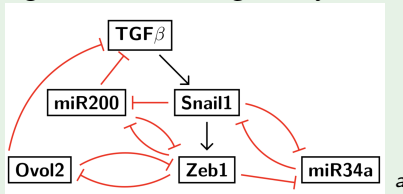
Examples of monoid labeled graphs and monoid graded categories

Example

For $M = 1$, any graph becomes a 1-graph in a unique way, by labeling each edge with 1, and this gives an isomorphism of categories $1\text{-Gph} \cong \text{Gph}$.

Example

For $M = (\mathbb{Z}/2, \cdot)$, the multiplicative monoid of the two element commutative ring, we get a standard *regulatory network*:



It is typical to represent elements of $(\mathbb{Z}/2, \cdot)$ as usual arrows (activation '+') and flat-head arrows (inhibitions '-')

$\cdot$	+	-
+	+	-
-	-	+

^aA 6-node EMT network, taken from Ying Xin, Bree Cummins, and Tomas Gedeon. *Multistability in the epithelial-mesenchymal transition network*. In: BMC bioinformatics 21.1 (2020)

Example

For $M = (\mathbb{Z}/3, \cdot)$, the multiplicative monoid of $\mathbb{Z}/3$, if we write the elements as $\{+, 0, -1\}$, we get the following multiplication table:

$\cdot$	$+$	0	$-$
$+$	$+$	0	$-$
0	0	0	0
$-$	$-$	0	$+$

There are at least three distinct interpretations of $\{+, 0, -\}$ -graphs:

- 1 We can use $+$ to denote a positive effect, $-$ to denote a negative effect, and 0 to denote *an effect whose value varies depending on the circumstances*.
- 2 We can use $+$ to denote a positive effect, $-$ to denote a negative effect, and 0 to denote *an effect that we deem negligible*.
- 3 We can use $+$ to denote a positive effect, $-$ to denote a negative effect, and 0 to denote *an unknown effect*.

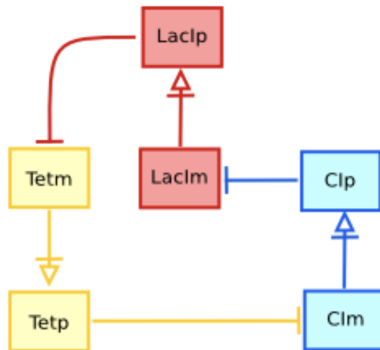
Example

For $M = (\mathbb{Z}/3 \cup \{I\}, \cdot)$, the four elements $\{I, +, 0, -\}$ monoid obtained from $\mathbb{Z}/3$ by adjoining a new identity element I , we get the following multiplication table:

$\cdot$	I	$+$	0	$-$
I	I	$+$	0	$-$
$+$	$+$	$+$	0	$-$
0	0	0	0	0
$-$	$-$	$-$	0	$+$

M -graphs can handle a large fraction of Systems Biology Graphical notations Activation Flow (SBGN-AF) diagrams, where

- instead of an edge labeled I , we write $\longrightarrow$ for 'necessary stimulation'.
- instead of an edge labeled $+$, we write $\longrightarrow$ for 'positive influence'.
- Instead of an edge labeled 0 , we write $\longrightarrow$ for 'unknown influence'.
- Instead of an edge labeled $-$, we write $\longrightarrow$ for 'negative influence'.



The SBGN-AF diagram representation of repressilator, often seen as one of the milestones in the birth of Synthetic Biology.

Example

Products of monoids are also useful: for example, to *describe both time delays and whether an effect is positive or negative*, we can use the monoid $([0, \infty), +, 0) \times \{+, -\}$.

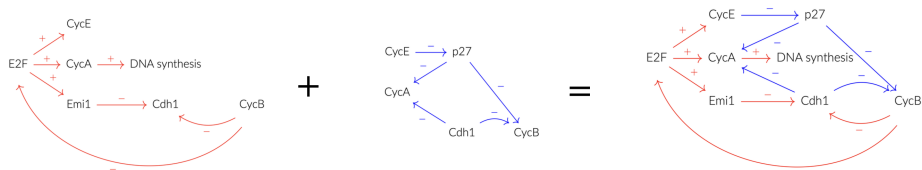
Example

The state transition graph of an asynchronous Boolean network $(V, \{F_v: \mathbb{B}^V \rightarrow \mathbb{B}\}_{v \in V})$ can be described as a $\text{Free}(V)$ -labeled graph, where $\text{Free}(V)$ is the free monoid generated by set of vertices V . A labeled edge $s_i \xrightarrow{\{v_1, v_2, \dots, v_k\}} s_j$ says that the state s_j is obtained from the state s_i by updating the states of the vertices $v_1, v_2, \dots, v_k$.

We believe by **suitably choosing the monoid M** , a M -labeled graph (G, ℓ) and the associated M -graded category $F_M(G, \ell)$ can be used to model various real world network systems, including of course various kinds of network representations of biological systems.

Next, we see an application of monoid graded categories in quantifying the ‘*degree of emergence*’ in an emergent feedback loop.

Composition of monoid labeled graphs and the degree of emergence in an emergent loop



For the monoid $M = (\mathbb{Z}/2, \cdot)$, let X be the red graph, Y be the blue graph, and $B := \{1, 2, 3, 4\}$.

Suppose

$$o: \text{disc}(B) \rightarrow X,$$

$$1 \mapsto \text{Cyc}E, 2 \mapsto \text{Cyc}A, 3 \mapsto \text{Cdh}1, 4 \mapsto \text{Cyc}B$$

and

$$i: \text{disc}(B) \rightarrow Y,$$

$$1 \mapsto \text{Cyc}E, 2 \mapsto \text{Cyc}A, 3 \mapsto \text{Cdh}1, 4 \mapsto \text{Cyc}B,$$

define the *interface points* along which red and blue graphs are composed.

Then, $X \cup Y := X +_{o, \text{disc}(B), i} Y$ is the *pushout* bicolored combined graph shown above consisting of both red and blue edges.

We would like to understand which of these paths are ‘emergent’, i.e., not already paths in X or Y .

Let $M\{x, y\}$ be the free monoid on two generators x and y . Then the category $\text{Free}(X \cup Y)$ has a unique $M\{x, y\}$ -grading where each edge of X has grade x and each edge of Y has grade y .

Let $I\{x, y\}$ be the quotient of the monoid $M\{x, y\}$ by relations saying that x and y are idempotent:

$$x^2 = x, \quad y^2 = y.$$

The quotient map $M\{x, y\} \rightarrow I\{x, y\}$ lets us push forward the $M\{x, y\}$ -grading on $\text{Free}(X \cup Y)$ to obtain an $I\{x, y\}$ -grading on this category.

This has a simplifying effect: Now the grade of a path records only how it goes back and forth between the graphs X and Y .

We would like to understand which of these paths are ‘emergent’, i.e., not already paths in X or Y .

Let $M\{x, y\}$ be the free monoid on two generators x and y . Then the category $\text{Free}(X \cup Y)$ has a unique $M\{x, y\}$ -grading where each edge of X has grade x and each edge of Y has grade y .

Let $I\{x, y\}$ be the quotient of the monoid $M\{x, y\}$ by relations saying that x and y are idempotent:

$$x^2 = x, \quad y^2 = y.$$

The quotient map $M\{x, y\} \rightarrow I\{x, y\}$ lets us push forward the $M\{x, y\}$ -grading on $\text{Free}(X \cup Y)$ to obtain an $I\{x, y\}$ -grading on this category.

This has a simplifying effect: Now the grade of a path records only how it goes back and forth between the graphs X and Y .

We would like to understand which of these paths are ‘emergent’, i.e., not already paths in X or Y .

Let $M\{x, y\}$ be the free monoid on two generators x and y . Then the category $\text{Free}(X \cup Y)$ has a unique $M\{x, y\}$ -grading where each edge of X has grade x and each edge of Y has grade y .

Let $I\{x, y\}$ be the quotient of the monoid $M\{x, y\}$ by relations saying that x and y are idempotent:

$$x^2 = x, \quad y^2 = y.$$

The quotient map $M\{x, y\} \rightarrow I\{x, y\}$ lets us push forward the $M\{x, y\}$ -grading on $\text{Free}(X \cup Y)$ to obtain an $I\{x, y\}$ -grading on this category.

This has a simplifying effect: Now the grade of a path records only how it goes back and forth between the graphs X and Y .

We would like to understand which of these paths are ‘emergent’, i.e., not already paths in X or Y .

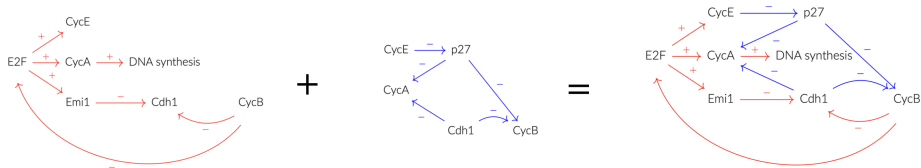
Let $M\{x, y\}$ be the free monoid on two generators x and y . Then the category $\text{Free}(X \cup Y)$ has a unique $M\{x, y\}$ -grading where each edge of X has grade x and each edge of Y has grade y .

Let $I\{x, y\}$ be the quotient of the monoid $M\{x, y\}$ by relations saying that x and y are idempotent:

$$x^2 = x, \quad y^2 = y.$$

The quotient map $M\{x, y\} \rightarrow I\{x, y\}$ lets us push forward the $M\{x, y\}$ -grading on $\text{Free}(X \cup Y)$ to obtain an $I\{x, y\}$ -grading on this category.

This has a simplifying effect: Now the grade of a path records only how it goes back and forth between the graphs X and Y .



If we apply the above grading to our bicolored combined graph $X \cup Y$ illustrated above, we get the grading of $Cdh1 \xrightarrow{-} CycB \xrightarrow{-} Cdh1$ as xy and the grading of $E2F \xrightarrow{+} CycE \xrightarrow{-} p27 \xrightarrow{-} CycB \xrightarrow{-} E2F$ as xyx .

This grading gives us a measure of the ‘amount of emergence’ involved in these loops arising as a consequence of composing the red and the blue graph along the nodes CycE, CycA, Cdh1 and CycB.

A remark on the composition of monoid labeled graphs

Remark

The above construction (via pushout) is one of the possible ways one can combine monoid labeled graphs into a composite one. We developed various algebraic ways to combine

- *a collection of monoid labeled graphs into a composite one*
- *to combine a collection of transformations (morphisms) of monoid labeled graphs into a composite transformation (morphism).*

*within a rigorous framework using the general theory of structured cospans^a in the setting of **symmetric monoidal double categories**. Details can be found in the Section 7 our paper.^b*

^aStructured Cospans, John C. Baez, Kenny Courser, Theory and Applications of Categories, Vol. 35, 2020, No. 48, pp 1771-1822

^bMotifs and Emergent Feedback in Labeled Graphs, arXiv:2506.23375, John C. Baez and Aditty Chaudhuri

Homology: An algebraic way to look at directed loops

Roughly speaking, homology of a space encodes the information of holes in a space via algebraic gadgets eg: abelian groups, etc.

When applied to graphs, homology captures ‘cycles’ present in a graph. Standard graph homology encodes the information of undirected cycles of the graph using *abelian groups*, especially the group of integers $\mathbb{Z}$.

We develop a homology for directed graphs (generalising the usual undirected graph homology) which encodes the information of directed cycles in the graph using commutative monoids.

Our homology can distinguish between these two:

$$G = u \begin{array}{c} \xrightarrow{\quad} \\ \xleftarrow{\quad} \end{array} v, \quad G' = u \begin{array}{c} \xrightarrow{\quad} \\ \xrightarrow{\quad} \end{array} v$$

Homology: An algebraic way to look at directed loops

Roughly speaking, homology of a space encodes the information of holes in a space via algebraic gadgets eg: abelian groups, etc.

When applied to graphs, homology captures ‘cycles’ present in a graph. Standard graph homology encodes the information of undirected cycles of the graph using *abelian groups*, especially the group of integers $\mathbb{Z}$.

We develop a homology for directed graphs (generalising the usual undirected graph homology) which encodes the information of directed cycles in the graph using commutative monoids.

Our homology can distinguish between these two:

$$G = u \begin{array}{c} \xrightarrow{\quad} \\ \xleftarrow{\quad} \end{array} v, \quad G' = u \begin{array}{c} \xrightarrow{\quad} \\ \xrightarrow{\quad} \end{array} v$$

Homology: An algebraic way to look at directed loops

Roughly speaking, homology of a space encodes the information of holes in a space via algebraic gadgets eg: abelian groups, etc.

When applied to graphs, homology captures ‘cycles’ present in a graph. Standard graph homology encodes the information of undirected cycles of the graph using *abelian groups*, especially the group of integers $\mathbb{Z}$.

We develop a homology for directed graphs (generalising the usual undirected graph homology) which encodes the information of directed cycles in the graph using commutative monoids.

Our homology can distinguish between these two:

$$G = u \begin{array}{c} \xrightarrow{\quad} \\ \xleftarrow{\quad} \end{array} v, \quad G' = u \begin{array}{c} \xrightarrow{\quad} \\ \xrightarrow{\quad} \end{array} v$$

A graph homology with coefficients in a commutative monoid

For any graph $G = (s, t: E \rightarrow V)$ and commutative monoid C , we consider

$$C_1(G, C) \begin{array}{c} \xrightarrow{C[s]} \\ \xrightarrow{C[t]} \end{array} C_0(G, C),$$

where

$$C_0(G, C) = C[V] = \left\{ \sum_{v \in V} a_v x \mid a_x \in C, \text{ all but finitely many } a_v \text{ are zero} \right\}$$

and

$$C_1(G, C) = C[E] = \left\{ \sum_{e \in E} a_e e \mid a_e \in C, \text{ all but finitely many } a_e \text{ are zero} \right\},$$

the monoid homomorphisms $C[s]$ and $C[t]$ are naturally induced respectively, by s and t .

We call $C_0(G, C)$ (resp. $C_1(G, C)$) the commutative monoid of **0-chains** (resp. **1-chains**) on G with coefficients in C . Taking the equalizer and coequalizer of $C[s]$ and $C[t]$ in the category of commutative monoids, we get

$$H_1(G, C) \xrightarrow{i} C_1(G, C) \begin{array}{c} \xrightarrow{C[s]} \\ \xrightarrow{C[t]} \end{array} C_0(G, C) \xrightarrow{p} H_0(G, C),$$

where we call the equalizer $H_1(G, C)$ and the coequalizer $H_0(G, C)$ as the **first** and **zeroth homology of G with coefficients in C** .

Concretely,

$$H_1(G, C) = \{x \in C_1(G, C) : C[s](x) = C[t](x)\} \underbrace{\leq}_{\text{submonoid}} C_1(G, C)$$

$$H_0(G, C) \cong C[\{\text{the set of undirected components of } G\}].$$

Relation to the homology with coefficients in an abelian group:

When C is an abelian group, we can rewrite

$$H_1(G, C) \xrightarrow{i} C_1(G, C) \xrightleftharpoons[C[t]]{C[s]} C_0(G, C) \xrightarrow{p} H_0(G, C),$$

as an exact sequence

$$0 \rightarrow H_1(G, C) \xrightarrow{i} C_1(G, C) \xrightarrow{d} C_0(G, C) \xrightarrow{p} H_0(G, C) \rightarrow 0,$$

which coincides with standard undirected graph homology when $C = \mathbb{Z}$, where

$$d = C[s] - C[t],$$

$$H_1(G, C) = \ker d, \quad H_0(G, C) = \operatorname{coker} d.$$

Directed loops as elements of $H_1(G, \mathbb{N})$

We call a **path**

$$\gamma = (v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_n)$$

in a graph G to be a **loop**, when $v_0 = v_n$. We call the loop γ to be **simple** if all the vertices $v_0, \dots, v_{n-1}$ are distinct.

Coefficients in $\mathbb{N}$:

For any path

$$\gamma = (v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_0)$$

in a graph G , we define $[\gamma] := e_1 + e_2 + \cdots + e_n \in C_1(G, \mathbb{N})$.

One can show that if γ is a loop, then $[\gamma] \in H_1(G, \mathbb{N}) \leq C_1(G, \mathbb{N})$.

We say two loops γ and δ **homologous** if $[\gamma] = [\delta]$, and call $[\gamma] \in H_1(G, \mathbb{N})$, the **homology class** of γ and δ .

Theorem (Baez-Chaudhuri)

For any graph G , the commutative monoid $H_1(G, \mathbb{N})$ is generated by the homology classes of simple loops in G . Every loop homologous to a simple loop

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_0$$

is of the form

$$v_k \xrightarrow{e_{k+1}} v_{k+1} \xrightarrow{e_{k+2}} \cdots \xrightarrow{e_{n+k-1}} v_{n+k-1} \xrightarrow{e_{n+k}} v_k$$

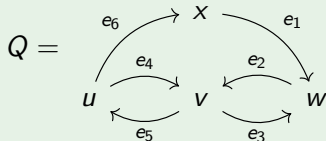
where we treat the subscripts as elements of $\mathbb{Z}/n$ and do addition mod n

Example

$$G = u \begin{array}{c} \xrightarrow{\quad} \\ \xleftarrow{\quad} \end{array} v, \quad G' = u \begin{array}{c} \xrightarrow{\quad} \\ \xrightarrow{\quad} \end{array} v$$

$H_1(G, \mathbb{N}) \cong \mathbb{N}$, while $H_1(G', \mathbb{N}) \cong \{0\}$. On the other hand, we have $H_1(G, \mathbb{Z}) \cong H_1(G', \mathbb{Z}) \cong \mathbb{Z}$.

Example

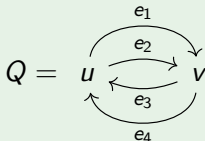


Consider the loop $c = e_1 + e_2 + e_3 + e_2 + e_5 + e_4 + e_5 + e_6$.

Observe that

$$c = (e_2 + e_3) + (e_4 + e_5) + (e_6 + e_1 + e_2 + e_5).$$

Example



Consider the loop $c = e_1 + e_3 + e_2 + e_4 \in H_1(Q, \mathbb{N})$.

Observe that

$$c = (e_1 + e_3) + (e_2 + e_4) = (e_1 + e_4) + (e_2 + e_3).$$

Feedback around a loop as a map on $H_1(G, \mathbb{N})$

For any commutative monoid M , and any M -labeled graph $(G, \ell: E \rightarrow M)$, there is a map

$$C_1(G, \mathbb{N}) \rightarrow M$$

defined by

$$\sum_{e \in E} n_e e \mapsto \sum_{e \in E} n_e \ell(e),$$

which restricts to a homomorphism of commutative monoids that we call $\tilde{\ell}: H_1(G, \mathbb{N}) \rightarrow M$.

Given a loop γ , we call $\tilde{\ell}([\gamma])$ **the feedback of (G, ℓ) around the loop γ** .

What does the homological formulation of feedback loops buys us?

We can now *encode the structural information of emergent feedback loops* algebraically using the language of **Mayer-Vietoris exact sequence** in the homology theory.

However, for doing this, we had to *invent* a version of Mayer-Vietoris exact sequence which *fits our homology theory*.

What does the homological formulation of feedback loops buys us?

We can now *encode the structural information of emergent feedback loops* algebraically using the language of **Mayer-Vietoris exact sequence** in the homology theory.

However, for doing this, we had to *invent* a version of Mayer-Vietoris exact sequence which *fits our homology theory*.

Emergent directed loops via Mayer-Vietoris

We proved the following version of Mayer–Vietoris for directed graph homology.

Theorem (Baez-Chaudhuri)

If C is a cancellative commutative monoid and X, Y are subgraphs of a graph $X \cup Y$ whose intersection $X \cap Y$ is a discrete graph, then the following is an equalizer diagram in the category of commutative monoids:

$$H_1(X, C) \oplus H_1(Y, C) \xrightarrow{\iota} H_1(X \cup Y, C) \begin{array}{c} \xrightarrow{qC[s]p_X} \\ \xrightarrow{qC[t]p_X} \end{array} H_0(X \cap Y, C),$$

where

- $\iota: H_1(X, C) \oplus H_1(Y, C) \rightarrow H_1(X \cup Y, C), (c_X, c_Y) \mapsto c_X + c_Y.$

Theorem (continued)

- $p_X: C_1(X \cup Y, C) \rightarrow C_1(X, C)$ by

$$p_X(c) = c_X,$$

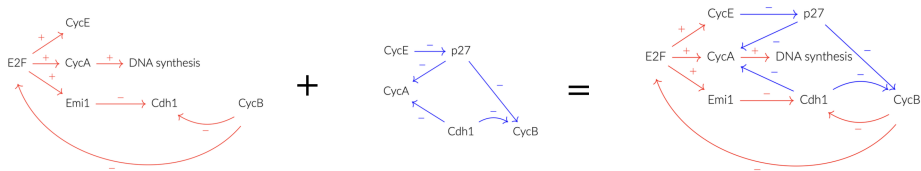
- $C[s], C[t]: C_1(X, C) \rightarrow C_0(X, C)$ induced from source and target maps,
- $q: C_0(X, C) \rightarrow H_0(X \cap Y, C)$ induced by

$$q': C_0(X, C) \rightarrow C_0(X \cap Y, C),$$

given as

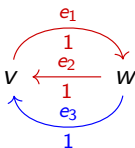
$$q' \left(\sum_{v \in \{\text{vertices of } X\}} c_v v \right) = \sum_{v \in \{\text{vertices of } X \cap Y\}} c_v v.$$

By the symmetry between X and Y , the analogous diagram with p_X replaced by p_Y is also an equalizer.



In the context of our example, the associated monomorphism $\iota: H_1(X, \mathbb{N}) \oplus H_1(Y, \mathbb{N}) \rightarrow H_1(X \cup Y, \mathbb{N})$, $(c_X, c_Y) \mapsto c_X + c_Y$ is not an isomorphism. Observe that the feedback around the emergent feedback loops $Cdh1 \xrightarrow{-} CycB \xrightarrow{-} Cdh1$ is '+', and along $E2F \xrightarrow{+} CycE \xrightarrow{-} p27 \xrightarrow{-} CycB \xrightarrow{-} E2F$ is '-', as the above graph is a $\mathbb{Z}/2$ -labeled graph.

If C is not cancellative, it is possible to have a cycle $c_X \in H_1(X, C)$ and a 1-chain that is not a cycle, $c_Y \in C_1(Y, C)$, whose sum is a cycle $c_X + c_Y \in H_1(X \cup Y, C)$.



Above is a $\mathbb{B}$ -labeled graph where $\mathbb{B} = \{0, 1\}$ is the additive Boolean monoid with addition as 'OR'. The subgraph X contains only the red edges, while the subgraph Y contains only the blue edges. The vertices lie in $X \cap Y$, which is the discrete graph on these two vertices. Let

$$c_X = e_1 + e_2, \quad c_Y = e_3.$$

Then c_X is a cycle, c_Y is not a cycle.

However, $c_X + c_Y$ is a cycle because

$$s(c_X + c_Y) = s(e_1 + e_2 + e_3) = v + w + w = v + w,$$

and

$$t(c_X + c_Y) = t(e_1 + e_2 + e_3) = w + v + v = v + w.$$

Future Research Directions

Our framework at its current stage has not explored any dynamical aspects. It may be interesting

- to investigate the dynamics (especially discrete dynamical models) that our framework can support,
- to investigate possible relations between our homology of graphs and the attractors of the associated discrete dynamical models,
- to decode the role of emergent feedback loops on such dynamics,
- to investigate the biological significance of our studies in the concrete examples of cross-talk phenomena between biochemical pathways.