Computing Generalized Rank Via Zigzag Persistence and its Applications

Woojin Kim

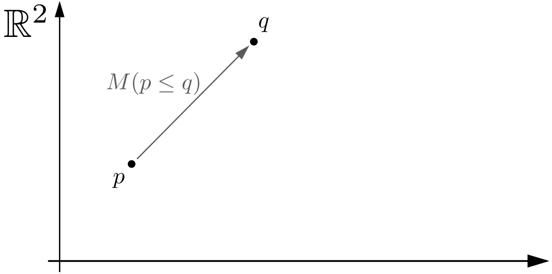
with Tamal Dey and Facundo Mémoli

Duke University
Department of Mathematics

June/09/2022

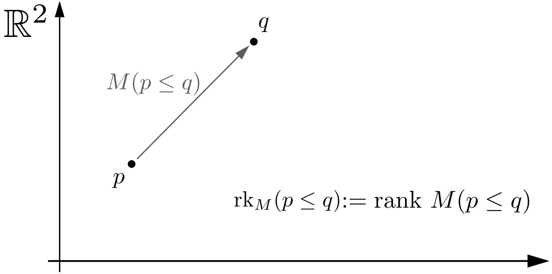
SoCG 2022, Berlin

2-parameter persistence and rank invariant



$M : \mathbb{R}^2 \to \textbf{vec}$

2-parameter persistence and rank invariant

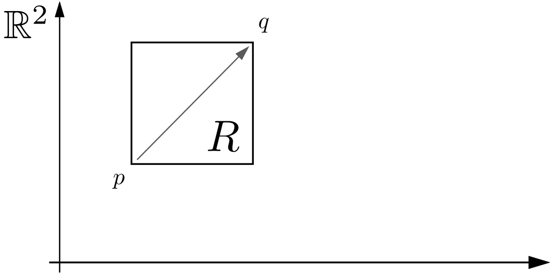


$M : \mathbb{R}^2 \to \mathbf{vec}$

Rank invariant [Carlsson, Zomordian]

$(p \leq q) \mapsto \mathbf{rk}_M(p \leq q)$

2-parameter persistence and rank invariant



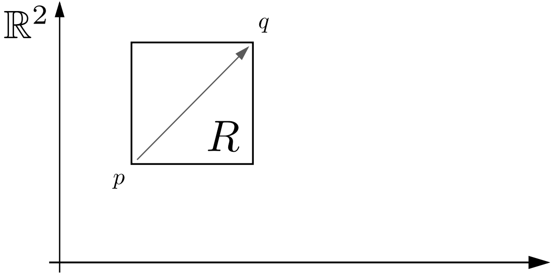
$M : \mathbb{R}^2 \to \textbf{vec}$

Rank invariant [Carlsson, Zomordian]

$(p \leq q) \mapsto \mathbf{rk}_M(p \leq q)$

$R \mapsto \mathbf{rk}_M(R)$

2-parameter persistence and rank invariant



$M : \mathbb{R}^2 \to \mathbf{vec}$

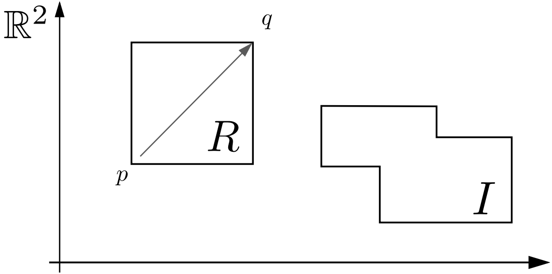
Rank invariant [Carlsson, Zomordian]

$(p \leq q) \mapsto \mathbf{rk}_M(p \leq q)$

$R \mapsto \mathbf{rk}_M(R)$

- A complete invariant of rectangle decomp. modules. [Botnan, Lebovici, Oudot]

2-parameter persistence and rank invariant



$M : \mathbb{R}^2 \to \mathbf{vec}$

Rank invariant [Carlsson, Zomordian]

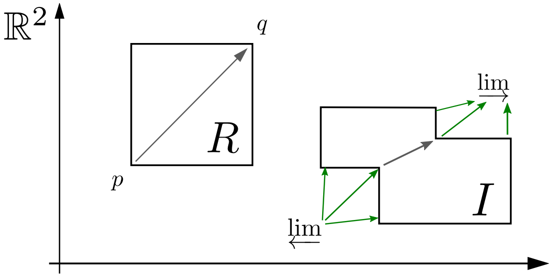
$(p \leq q) \mapsto \mathbf{rk}_M(p \leq q)$

$R \mapsto \mathbf{rk}_M(R)$

- A complete invariant of rectangle decomp. modules. [Botnan, Lebovici, Oudot]

Generalized rank invariant [K, Mémoli]

2-parameter persistence and rank invariant



$M : \mathbb{R}^2 \to \mathbf{vec}$

$\mathbf{rk}_M(I)$
$= \mathbf{rank}\left(\varprojlim M|_I \to \varinjlim M|_I\right)$

Rank invariant [Carlsson, Zomordian]

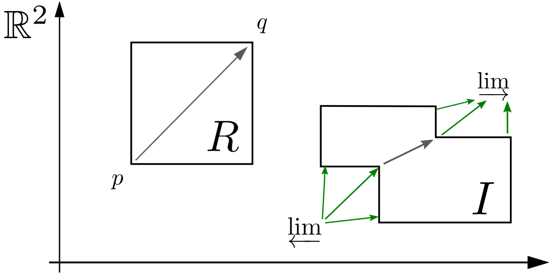
$(p \leq q) \mapsto \mathbf{rk}_M(p \leq q)$
$R \mapsto \mathbf{rk}_M(R)$

- A complete invariant of rectangle decomp. modules. [Botnan, Lebovici, Oudot]

Generalized rank invariant [K, Mémoli]

$I \mapsto \mathbf{rk}_M(I)$

2-parameter persistence and rank invariant



$M : \mathbb{R}^2 \to \mathbf{vec}$

$\mathbf{rk}_M(I)$

$= \mathbf{rank}\left(\varprojlim M|_I \to \varinjlim M|_I\right)$

Rank invariant [Carlsson, Zomordian]

$(p \leq q) \mapsto \mathbf{rk}_M(p \leq q)$

$R \mapsto \mathbf{rk}_M(R)$

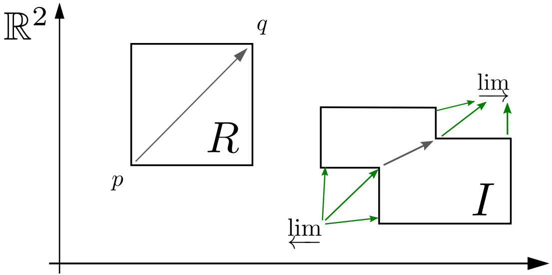
- A complete invariant of rectangle decomp. modules. [Botnan, Lebovici, Oudot]

Generalized rank invariant [K, Mémoli]

$I \mapsto \mathbf{rk}_M(I)$

- A complete invariant of int. decomp. modules. [K, Mémoli]

2-parameter persistence and rank invariant



$M : \mathbb{R}^2 \to \textbf{vec}$

$\textbf{rk}_M(I)$

$= \textbf{rank}\left(\varprojlim M|_I \to \varinjlim M|_I\right)$

Rank invariant [Carlsson, Zomordian]

$(p \leq q) \mapsto \textbf{rk}_M(p \leq q)$

$R \mapsto \textbf{rk}_M(R)$

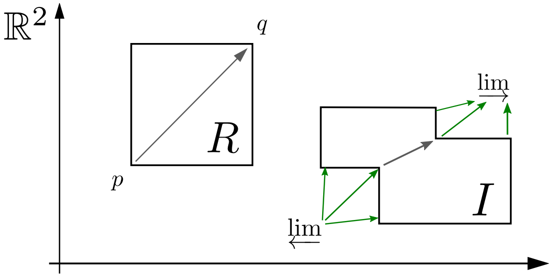
- A complete invariant of rectangle decomp. modules. [Botnan, Lebovici, Oudot]

Generalized rank invariant [K, Mémoli]

$I \mapsto \textbf{rk}_M(I)$

- A complete invariant of int. decomp. modules. [K, Mémoli]
- Stable. [K, Mémoli]

2-parameter persistence and rank invariant



$M : \mathbb{R}^2 \to \mathbf{vec}$

$\mathbf{rk}_M(I)$
$= \mathbf{rank}\left(\varprojlim M|_I \to \varinjlim M|_I\right)$

Rank invariant [Carlsson, Zomordian]

$(p \leq q) \mapsto \mathbf{rk}_M(p \leq q)$

$R \mapsto \mathbf{rk}_M(R)$

- A complete invariant of rectangle decomp. modules. [Botnan, Lebovici, Oudot]

Generalized rank invariant [K, Mémoli]

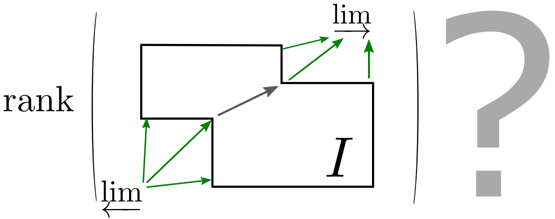
$I \mapsto \mathbf{rk}_M(I)$

- A complete invariant of int. decomp. modules. [K, Mémoli]
- Stable. [K, Mémoli]
- Möbius $\rightsquigarrow$ Bigraded Betti numbers. [K, Moore]

How to compute the generalized rank invariant?

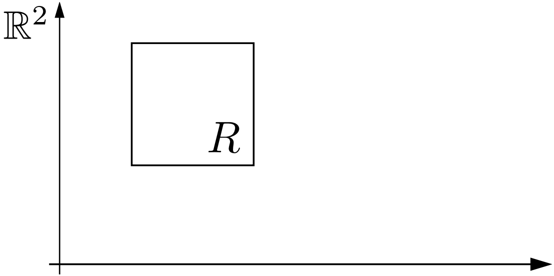
How to compute the generalized rank invariant?

How to compute



Main theorem

$M : \mathbb{R}^2 \to \mathbf{vec}$

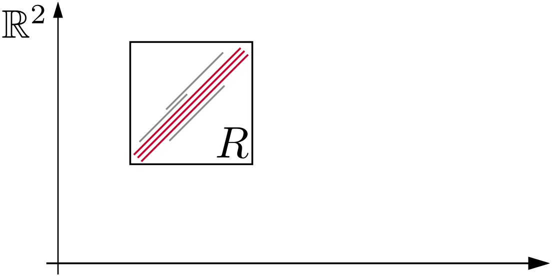


Rank invariant

$R \mapsto \mathbf{rk}_M(R)$

Main theorem

$M : \mathbb{R}^2 \to \textbf{vec}$

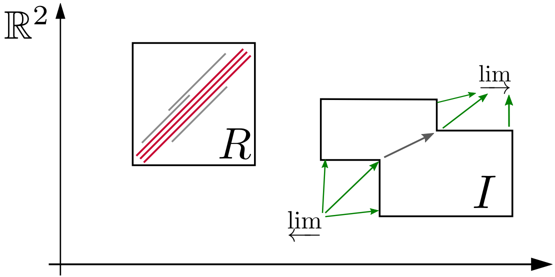


Rank invariant (known)

$$R \mapsto \textbf{rk}_M(R)$$
$$= \#(\text{“Full” bars})$$

Main theorem

$M : \mathbb{R}^2 \to \textbf{vec}$



Rank invariant (known)

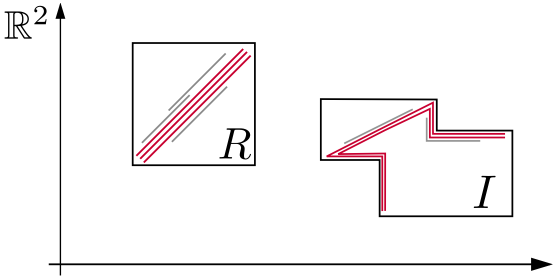
$R \mapsto \mathbf{rk}_M(R)$

$= \#($"Full" bars$)$

Theorem 1 [Dey, K, Mémoli]

$I \mapsto \mathbf{rk}_M(I)$

Main theorem



$M : \mathbb{R}^2 \to \textbf{vec}$

Rank invariant (known)

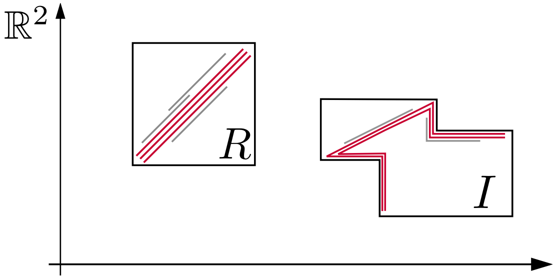
$$R \mapsto \textbf{rk}_M(R) = \#(\text{“Full” bars})$$

Theorem 1 [Dey, K, Mémoli]

$$I \mapsto \textbf{rk}_M(I) = \#(\text{“Full” bars})$$

Main theorem

$M : \mathbb{R}^2 \to \mathbf{vec}$

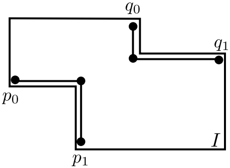


Rank invariant (known)

$$R \mapsto \mathbf{rk}_M(R) = \#(\text{"Full" bars})$$

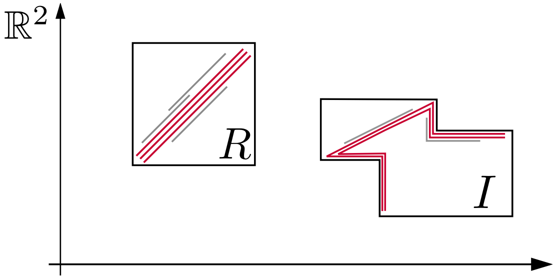
Theorem 1 [Dey, K, Mémoli]

$$I \mapsto \mathbf{rk}_M(I) = \#(\text{"Full" bars})$$



Main theorem

$M : \mathbb{R}^2 \to \textbf{vec}$

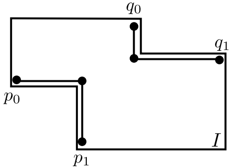


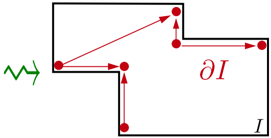
Rank invariant (known)

$$R \mapsto \textbf{rk}_M(R)$$
$$= \#(\text{"Full" bars})$$

Theorem 1 [Dey, K, Mémoli]

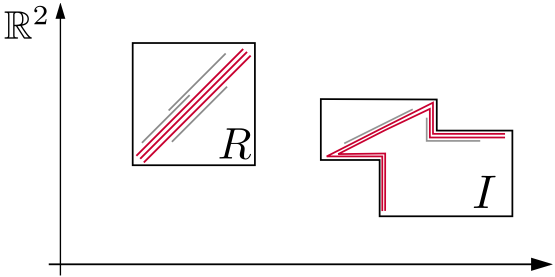
$$I \mapsto \textbf{rk}_M(I)$$
$$= \#(\text{"Full" bars})$$





Main theorem

$M : \mathbb{R}^2 \to \textbf{vec}$

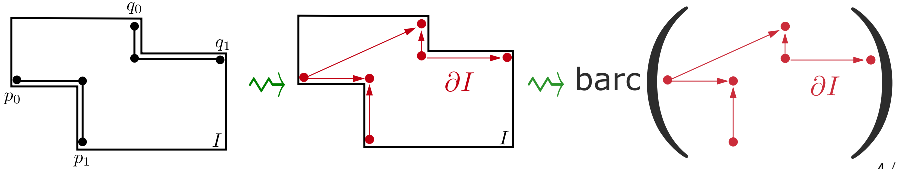


Rank invariant (known)

$R \mapsto \mathbf{rk}_M(R)$
$= \#(\text{“Full” bars})$

Theorem 1 [Dey, K, Mémoli]

$I \mapsto \mathbf{rk}_M(I)$
$= \#(\text{“Full” bars})$



$\mathbb{P}$: A partially ordered set (poset), as a category. $p \leq q \Leftrightarrow p \xrightarrow{\exists !} q$.

$\mathbb{P}$: A partially ordered set (poset), as a category. $p \leq q \Leftrightarrow p \xrightarrow{\exists!} q$.

vec : The category of finite dimensional vector spaces over a field $\mathbb{F}$.

$\mathbb{P}$: A partially ordered set (poset), as a category. $p \leq q \Leftrightarrow p \xrightarrow{\exists !} q$.

vec : The category of finite dimensional vector spaces over a field $\mathbb{F}$.

A functor $M : \mathbb{P} \to$ **vec** is called a **($\mathbb{P}$-indexed) persistence module.**

$\mathbb{P}$: A partially ordered set (poset), as a category. $p \leq q \Leftrightarrow p \xrightarrow{\exists!} q$.

vec : The category of finite dimensional vector spaces over a field $\mathbb{F}$.

A functor $M : \mathbb{P} \to$ **vec** is called a **($\mathbb{P}$-indexed) persistence module.**

$\mathbb{P}$ will be finite, unless stated otherwise.

$\mathbb{P}$: A partially ordered set (poset), as a category. $p \leq q \Leftrightarrow p \overset{\exists!}{\to} q$.

vec : The category of finite dimensional vector spaces over a field $\mathbb{F}$.

A functor $M : \mathbb{P} \to$ **vec** is called a **($\mathbb{P}$-indexed) persistence module.**

$\mathbb{P}$ will be finite, unless stated otherwise.

1-parameter

- a vertex $\leftarrow$ a vector space
- an arrow $\leftarrow$ a linear map

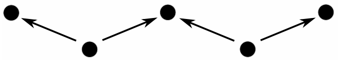
$\mathbb{P}$: A partially ordered set (poset), as a category. $p \leq q \Leftrightarrow p \overset{\exists!}{\to} q$.

vec : The category of finite dimensional vector spaces over a field $\mathbb{F}$.

A functor $M : \mathbb{P} \to$ **vec** is called a **($\mathbb{P}$-indexed) persistence module.**

$\mathbb{P}$ will be finite, unless stated otherwise.

1-parameter



Zigzag

- a vertex $\leftarrow$ a vector space
- an arrow $\leftarrow$ a linear map

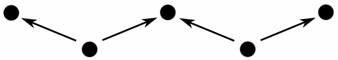
$\mathbb{P}$: A partially ordered set (poset), as a category. $p \leq q \Leftrightarrow p \overset{\exists!}{\to} q$.

vec : The category of finite dimensional vector spaces over a field $\mathbb{F}$.

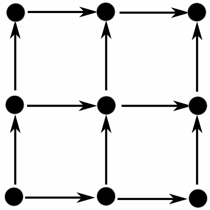
A functor $M : \mathbb{P} \to$ **vec** is called a **($\mathbb{P}$-indexed) persistence module.**

$\mathbb{P}$ will be finite, unless stated otherwise.

1-parameter



Zigzag



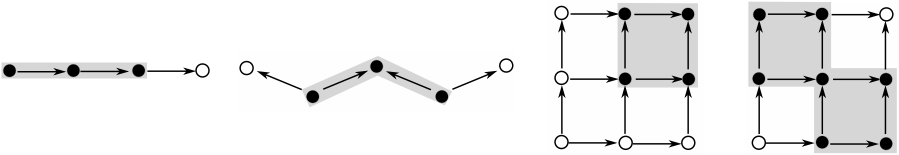
2-parameter

- a vertex $\leftarrow$ a vector space
- an arrow $\leftarrow$ a linear map
- Every closed subdiagram commutes.

- $I \subset \mathbb{P}$ is called an **interval** if I is **connected**

- $I \subset \mathbb{P}$ is called an **interval** if I is **connected**

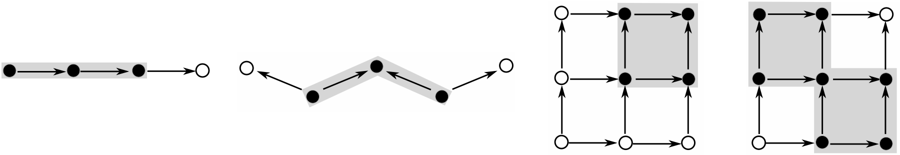
Examples.



- $I \subset \mathbb{P}$ is called an **interval** if I is **connected** and **order-convex**, i.e.

$$[p, q \in I \text{ and } p \leq r \leq q] \Rightarrow r \in I.$$

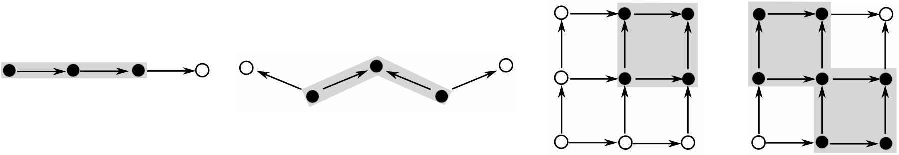
Examples.



- $I \subset \mathbb{P}$ is called an **interval** if I is **connected** and **order-convex**, i.e.

$$[p, q \in I \text{ and } p \le r \le q] \Rightarrow r \in I.$$

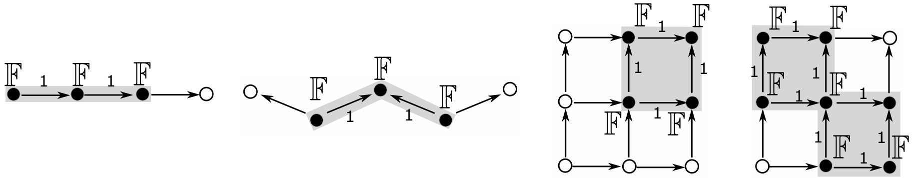
Examples.



- $I \subset \mathbb{P}$ is called an **interval** if I is **connected** and **order-convex**, i.e.

$$[p, q \in I \text{ and } p \leq r \leq q] \Rightarrow r \in I.$$

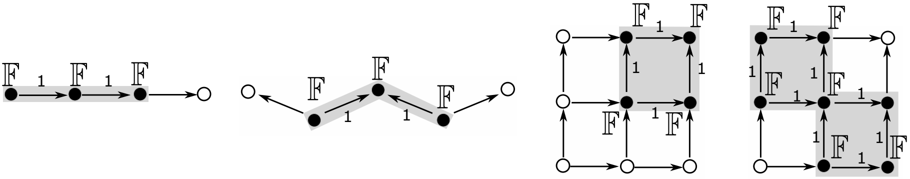
An **interval module** $\mathbb{I}^I : \mathbb{P} \to \textbf{vec}$ is an indicator module over I.



- $I \subset \mathbb{P}$ is called an **interval** if I is **connected** and **order-convex**, i.e.

$$[p, q \in I \text{ and } p \leq r \leq q] \Rightarrow r \in I.$$

An **interval module** $\mathbb{I}^I : \mathbb{P} \to \textbf{vec}$ is an indicator module over I.



Interval modules are **indecomposable**.

Definition

If $M \cong \bigoplus_i \mathbb{I}^{I_i}$, then M is called **interval decomposable**, and the multiset $\{\{I_i\}\}$ is called the **barcode** of M.

Rank over an interval $I \subset \mathbb{P}$

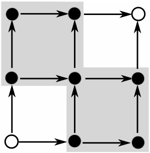
Given $M : \mathbb{P} \to \textbf{vec}$, and any interval $I \subset \mathbb{P}$

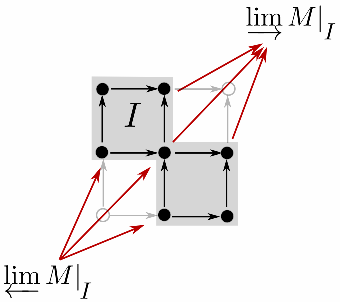
$$\mathbf{rk}_M(I) := \mathbf{rank}\left(\phi_M : \varprojlim M|_I \to \varinjlim M|_I\right)$$

Rank over an interval $I \subset \mathbb{P}$

Given $M : \mathbb{P} \to \textbf{vec}$, and any interval $I \subset \mathbb{P}$

$$\textbf{rk}_M(I) := \textbf{rank}\left(\phi_M : \varprojlim M|_I \to \varinjlim M|_I\right)$$





Int($\mathbb{P}$): the set of all intervals of $\mathbb{P}$.

(**Int**-)Generalized rank invariant of $M : \mathbb{P} \to \textbf{vec}$ [K,Mémoli], [Asashiba et al.]

$$\begin{aligned}\textbf{rk}_M : \textbf{Int}(\mathbb{P}) &\to \mathbb{Z}_{\geq 0}\\ I &\mapsto \textbf{rk}_M(I)\end{aligned}$$

Remarks.

$\mathbf{Int}(\mathbb{P})$: the set of all intervals of $\mathbb{P}$.

(**Int**-)Generalized rank invariant of $M : \mathbb{P} \to \mathbf{vec}$ [K,Mémoli], [Asashiba et al.]

$$\begin{aligned} \mathbf{rk}_M : \mathbf{Int}(\mathbb{P}) &\to \mathbb{Z}_{\geq 0} \\ I &\mapsto \mathbf{rk}_M(I) \end{aligned}$$

Remarks.

1. $\mathbf{rk}_M$ subsumes the standard rank invariant.

$[p, q]$

Int($\mathbb{P}$): the set of all intervals of $\mathbb{P}$.

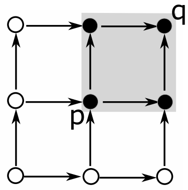
(Int-)Generalized rank invariant of $M : \mathbb{P} \to \mathbf{vec}$ [K,Mémoli], [Asashiba et al.]

$$\mathbf{rk}_M : \mathbf{Int}(\mathbb{P}) \to \mathbb{Z}_{\geq 0}$$
$$I \mapsto \mathbf{rk}_M(I)$$

Remarks.

1. $\mathbf{rk}_M$ subsumes the standard rank invariant.

$$[p,q] \mapsto \mathbf{rk}_M([p,q])$$
$$= \mathbf{rank}(M(p \leq q))$$



Int($\mathbb{P}$): the set of all intervals of $\mathbb{P}$.

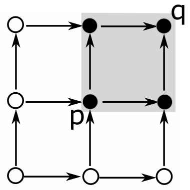
(**Int**-)Generalized rank invariant of $M : \mathbb{P} \to \textbf{vec}$ [K,Mémoli], [Asashiba et al.]

$$\textbf{rk}_M : \textbf{Int}(\mathbb{P}) \to \mathbb{Z}_{\geq 0}$$
$$I \mapsto \textbf{rk}_M(I)$$

Remarks.

1. $\textbf{rk}_M$ subsumes the standard rank invariant.

$$[p, q] \mapsto \textbf{rk}_M([p, q]) = \textbf{rank}(M(p \leq q))$$



2. $\textbf{rk}_M$ is non-increasing, i.e.

$$I \subset J \;\Rightarrow\; \textbf{rk}_M(I) \geq \textbf{rk}_M(J).$$

Suppose that M is interval decomposable.

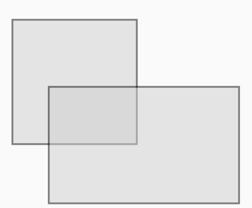
Proposition

$$\mathbf{rk}_M(I) = \# \{\!\{ J \in \mathbf{barc}(M) : I \subset J \}\!\} .$$

Suppose that M is interval decomposable.

Proposition

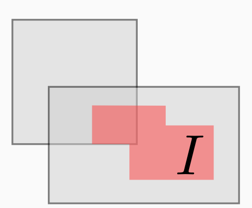
$\mathbf{rk}_M(I) = \# \{\!\{ J \in \mathbf{barc}(M) : I \subset J \}\!\}$.

$M \cong$ 

Suppose that M is interval decomposable.

Proposition

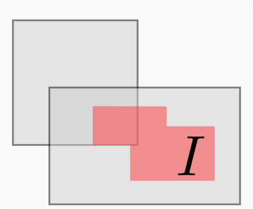
$\mathbf{rk}_M(I) = \# \{\!\{ J \in \mathbf{barc}(M) : I \subset J \}\!\}$.

$M \cong$ 

Suppose that M is interval decomposable.

Proposition

$\mathbf{rk}_M(I) = \#\,\{\!\{J \in \mathbf{barc}(M) : I \subset J\}\!\}\,.$

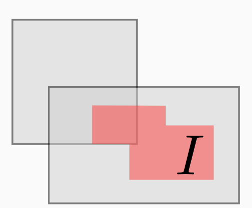
$M \cong$ 

$\mathbf{rk}_M(I) = 1.$

Suppose that M is interval decomposable.

Proposition

$\mathbf{rk}_M(I) = \# \{\!\{ J \in \mathbf{barc}(M) : I \subset J \}\!\}$.

$M \cong$ 

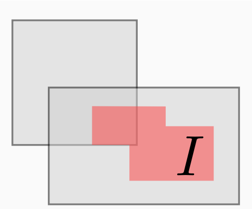
$\mathbf{rk}_M(I) = 1.$

Via the inclusion-exclusion principle (or Möbius inversion), the multiplicity of each interval in $\mathbf{barc}(M)$ can be computed. (i.e. $\mathbf{rk}_M$ is a complete inv. of int. decomp. modules.)

Suppose that M is interval decomposable.

Proposition

$\mathbf{rk}_M(I) = \#\{\!\{J \in \mathbf{barc}(M) : I \subset J\}\!\}.$

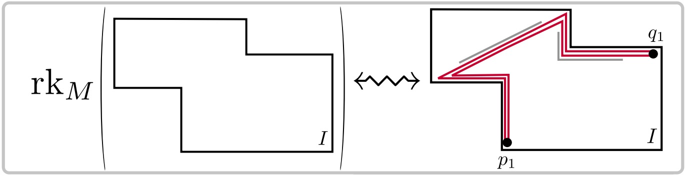
$M \cong$ 

$\mathbf{rk}_M(I) = 1.$

Via the inclusion-exclusion principle (or Möbius inversion), the multiplicity of each interval in $\mathbf{barc}(M)$ can be computed. (i.e. $\mathbf{rk}_M$ is a complete inv. of int. decomp. modules.)

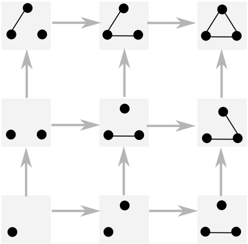
The Möbius inversion of $\mathbf{rk}_M$ yields:

1. Generalized persistence diagram of M. [K,Mémoli], [Patel]
2. Interval approximation of M. [Asashiba et al.]
3. Rank decomposition of M. [Botnan, Oppermann, Oudot]

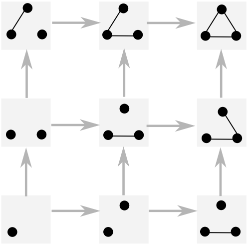


Applications

Input. A simplicial filtration $\mathcal{F}$ over a finite 2d-grid.

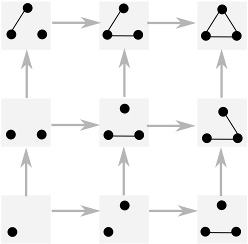


Input. A simplicial filtration $\mathcal{F}$ over a finite 2d-grid.



$M := \mathrm{H}(\mathcal{F})$

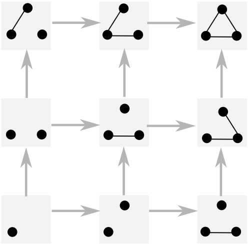
Input. A simplicial filtration $\mathcal{F}$ over a finite 2d-grid.



$M := \mathrm{H}(\mathcal{F})$

Algorithm 1. Given that M is interval decomposable, compute the barcode of M.

Input. A simplicial filtration $\mathcal{F}$ over a finite 2d-grid.



$M := \mathrm{H}(\mathcal{F})$

Algorithm 1. Given that M is interval decomposable, compute the barcode of M.

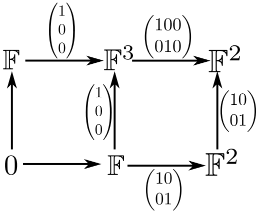
Algorithm 2. Check whether M is interval decomposable or not.

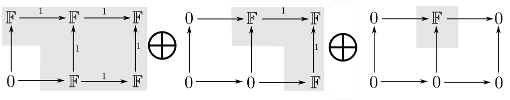
Algorithm 1. Compute the the barcode of M.

$$
M \;=\; \begin{array}{ccccc}
\mathbb{F} & \xrightarrow{\begin{pmatrix}1\\0\\0\end{pmatrix}} & \mathbb{F}^3 & \xrightarrow{\begin{pmatrix}100\\010\end{pmatrix}} & \mathbb{F}^2 \\
\uparrow & & \uparrow{\scriptstyle\begin{pmatrix}1\\0\\0\end{pmatrix}} & & \uparrow{\scriptstyle\begin{pmatrix}10\\01\end{pmatrix}} \\
0 & \longrightarrow & \mathbb{F} & \xrightarrow[\begin{pmatrix}10\\01\end{pmatrix}]{} & \mathbb{F}^2
\end{array}
$$

Algorithm 1. Compute the the barcode of M.

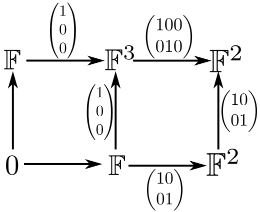
$M =$

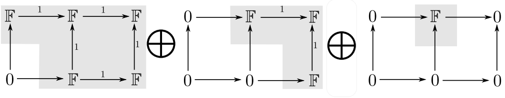




Algorithm 1. Compute the the barcode of M.

$M =$

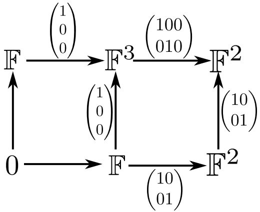


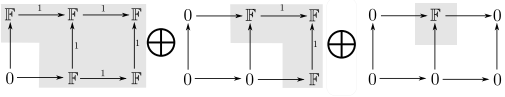


$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

Algorithm 1. Compute the the barcode of M.

$M =$ 



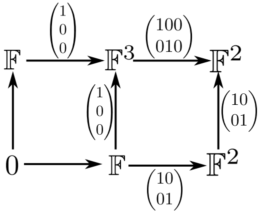
$\overrightarrow{\dim}(M) =$

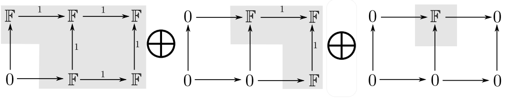
1	3	2
	1	2

Step 1. (Extension process until $\mathbf{rk}_M = 0$)

Algorithm 1. Compute the the barcode of M.

$M =$

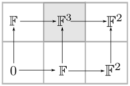




$\overrightarrow{\dim}(M) =$

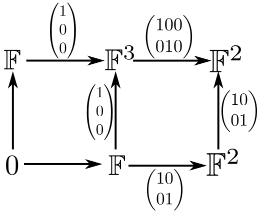
1	3	2
	1	2

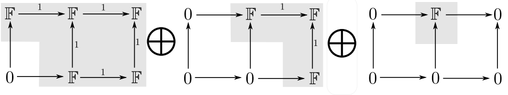
Step 1. (Extension process until $\mathbf{rk}_M = 0$)



$\mathbf{rk}_M(I) = 3$

Algorithm 1. Compute the the barcode of M.

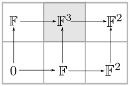
$M =$ 



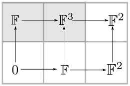
$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

Step 1. (Extension process until $\mathbf{rk}_M = 0$)

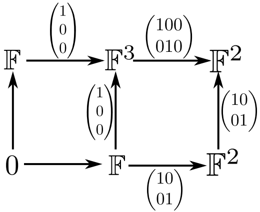


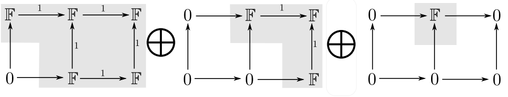
$\mathbf{rk}_M(I) = 3$



$\mathbf{rk}_M(I) = 1$

Algorithm 1. Compute the the barcode of M.

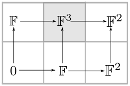
$M =$ 



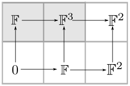
$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

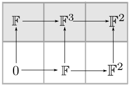
Step 1. (Extension process until $\mathbf{rk}_M = 0$)



$\mathbf{rk}_M(I) = 3$

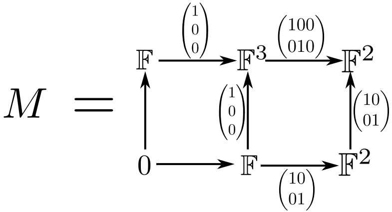


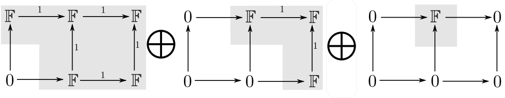
$\mathbf{rk}_M(I) = 1$



$\mathbf{rk}_M(I) = 1$

Algorithm 1. Compute the the barcode of M.

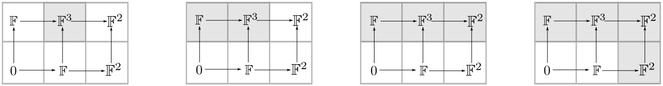




$\overrightarrow{\dim}(M) =$

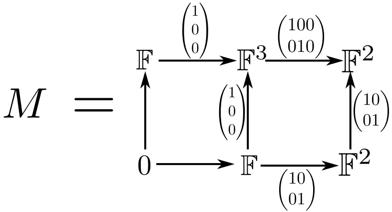
1	3	2
	1	2

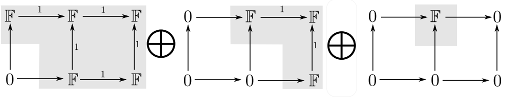
Step 1. (Extension process until $\mathbf{rk}_M = 0$)



$\mathbf{rk}_M(I) = 3$ $\quad$ $\mathbf{rk}_M(I) = 1$ $\quad$ $\mathbf{rk}_M(I) = 1$ $\quad$ $\mathbf{rk}_M(I) = 1$

Algorithm 1. Compute the the barcode of M.

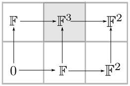




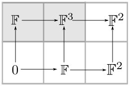
$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

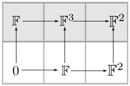
Step 1. (Extension process until $\mathbf{rk}_M = 0$)



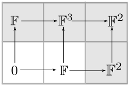
$\mathbf{rk}_M(I) = 3$



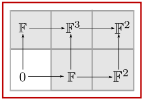
$\mathbf{rk}_M(I) = 1$



$\mathbf{rk}_M(I) = 1$

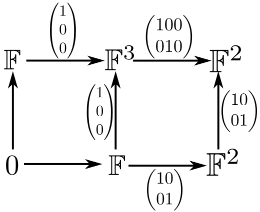


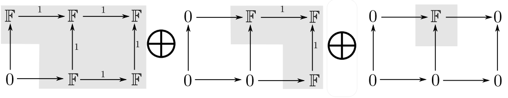
$\mathbf{rk}_M(I) = 1$



$\mathbf{rk}_M(I) = 1$

Algorithm 1. Compute the the barcode of M.

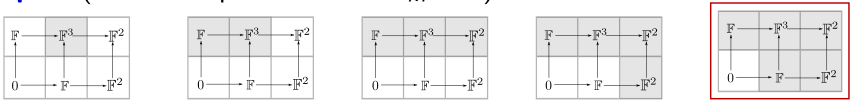
$M =$ 



$\overrightarrow{\dim}(M) =$

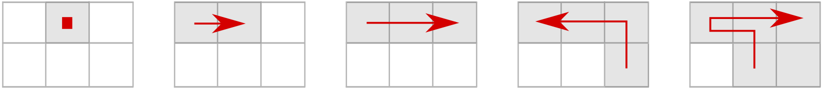
1	3	2
	1	2

Step 1. (Extension process until $\mathbf{rk}_M = 0$)



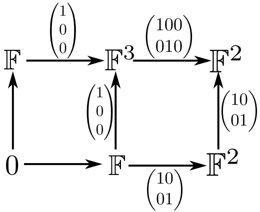
$\mathbf{rk}_M(I) = 3$ $\mathbf{rk}_M(I) = 1$ $\mathbf{rk}_M(I) = 1$ $\mathbf{rk}_M(I) = 1$ $\mathbf{rk}_M(I) = 1$

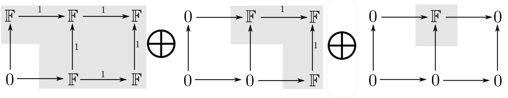
Remark. We will be using a zigzag persistence algorithm in this process.



[Milosavljevic, Morozov, Skraba 2011], [Dey, Tou 2022]

Algorithm 1. Compute the the barcode of M.

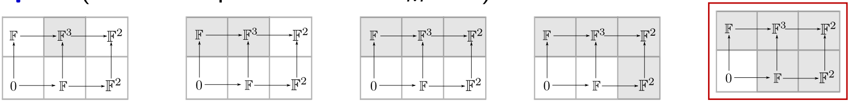
$M =$ 



$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

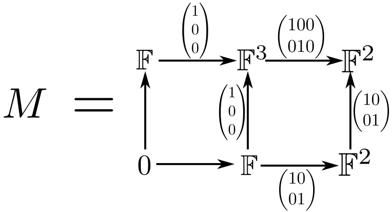
Step 1. (Extension process until $\mathbf{rk}_M = 0$)

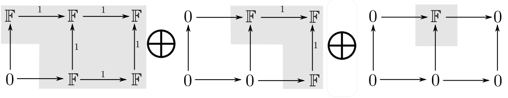


$\mathbf{rk}_M(I) = 1$

Step 2. (Subtraction)

Algorithm 1. Compute the the barcode of M.

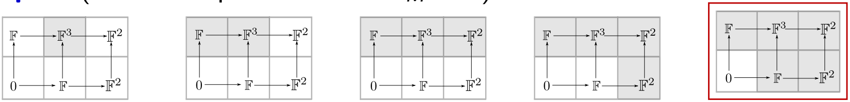




$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

Step 1. (Extension process until $\mathbf{rk}_M = 0$)



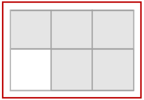
$\mathbf{rk}_M(I) = 1$

Step 2. (Subtraction)

$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

$-$ $1\bullet$

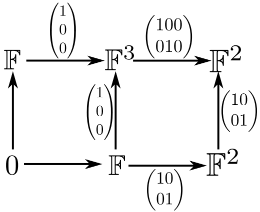


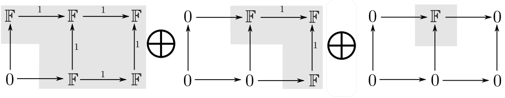
$=$

	2	1
		1

Algorithm 1. Compute the the barcode of M.

$M =$

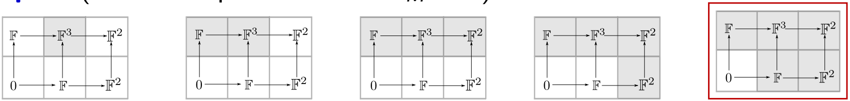




$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

Step 1. (Extension process until $\mathbf{rk}_M = 0$)



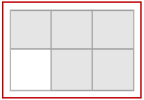
$\mathbf{rk}_M(I) = 1$

Step 2. (Subtraction)

$\overrightarrow{\dim}(M) =$

1	3	2
	1	2

$-\ 1\bullet$



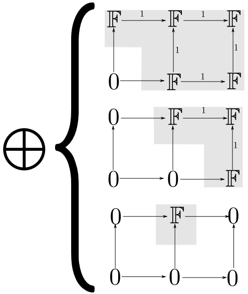
$=$

	2	1
		1

Repeat **Steps 1 and 2** until the dimension vector becomes zero.

One possible scenario:

$$M =$$



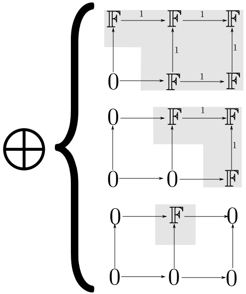
$\overrightarrow{\dim}(M)$

	2	1
		1

One possible scenario:

Step 1.

$M =$

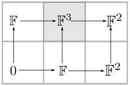


$\overrightarrow{\dim}(M)$

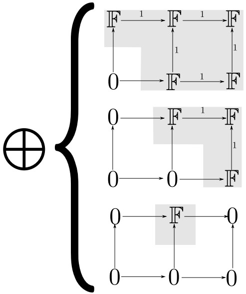
	2	1
		1

One possible scenario:

Step 1.



$$M =$$

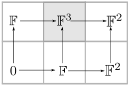


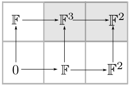
$\overrightarrow{\dim}(M)$

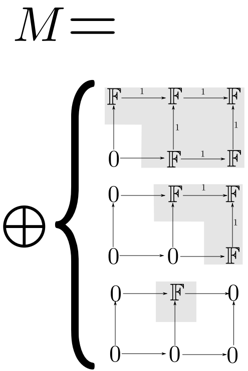
	2	1
		1

One possible scenario:

Step 1.





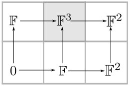


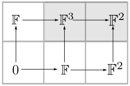
$\overrightarrow{\dim}(M)$

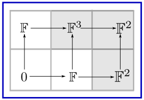
	2	1
		1

One possible scenario:

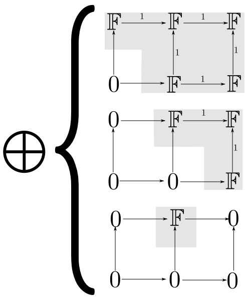
Step 1.







$M =$

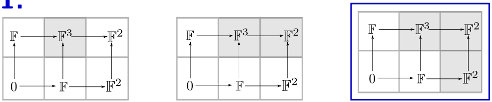


$\overrightarrow{\dim}(M)$

	2	1
		1

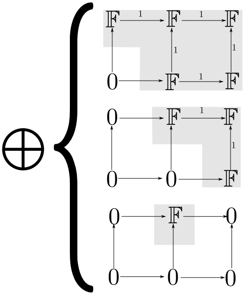
One possible scenario:

Step 1.



$\mathbf{rk}_M(I) = 2$

$M =$

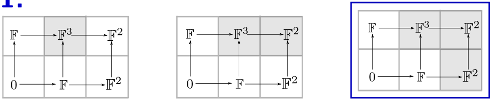


$\overrightarrow{\dim}(M)$

	2	1
		1

One possible scenario:

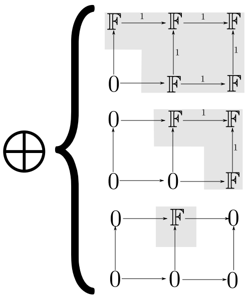
Step 1.



$\mathbf{rk}_M(I) = 2$

Step 2.

$M =$

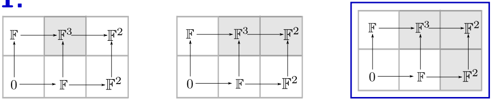


$\overrightarrow{\dim}(M)$

	2	1
		1

One possible scenario:

Step 1.

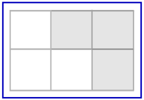


$\mathbf{rk}_M(I) = 2$

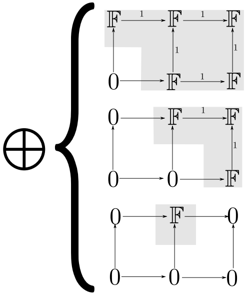
Step 2.

	2	1
		1

— 2•



$M = \oplus$

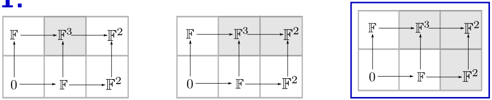


$\overrightarrow{\dim}(M)$

	2	1
		1

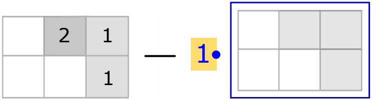
One possible scenario:

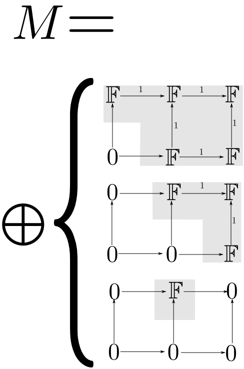
Step 1.



$\mathbf{rk}_M(I) = 2$

Step 2.



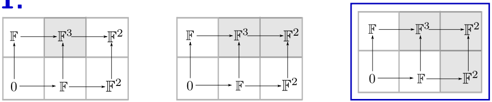


$\overrightarrow{\dim}(M)$

	2	1
		1

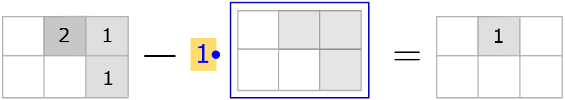
One possible scenario:

Step 1.

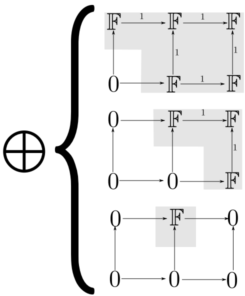


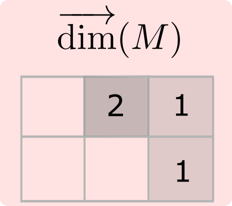
$\mathbf{rk}_M(I) = 2$

Step 2.



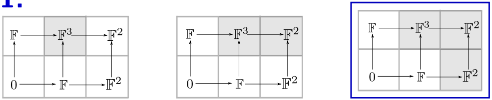
$M =$





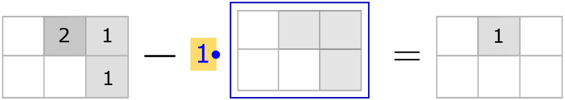
One possible scenario:

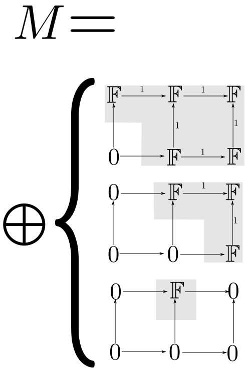
Step 1.

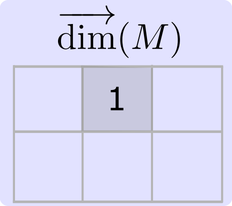


$\mathbf{rk}_M(I) = 2$

Step 2.

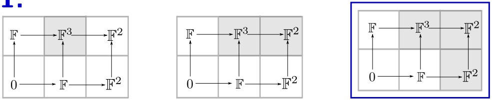






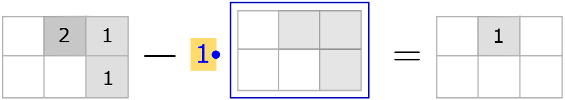
One possible scenario:

Step 1.

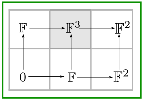


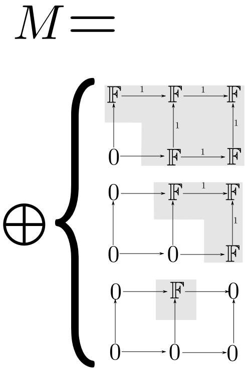
$\mathbf{rk}_M(I) = 2$

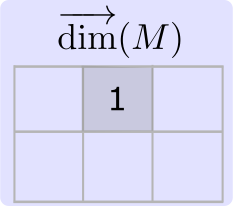
Step 2.



Step 1.

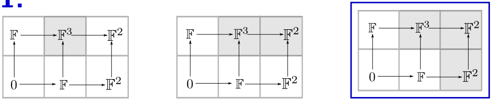






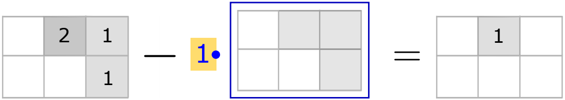
One possible scenario:

Step 1.

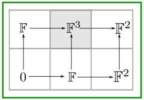


$\mathbf{rk}_M(I) = 2$

Step 2.



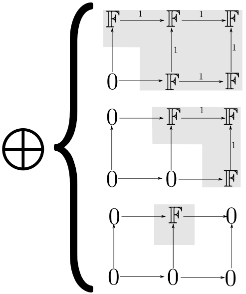
Step 1.



$\mathbf{rk}_M(I) = 3$

Step 2. $\overrightarrow{\dim}(M) - 1 \cdot \overrightarrow{\dim}(\mathbb{I}^I) = 0.$

$M =$

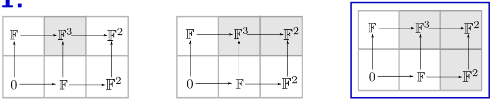


$\overrightarrow{\dim}(M)$

	1	

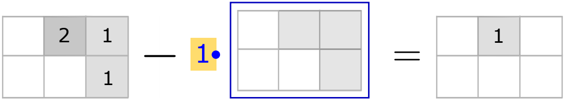
One possible scenario:

Step 1.

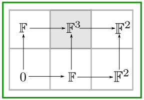


$\mathbf{rk}_M(I) = 2$

Step 2.



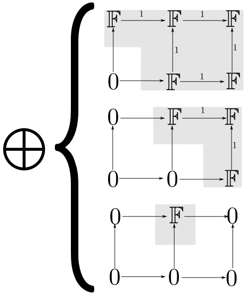
Step 1.

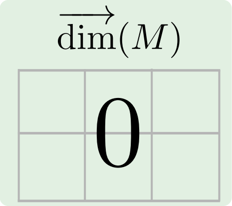


$\mathbf{rk}_M(I) = 3$

Step 2. $\overrightarrow{\dim}(M) - 1 \cdot \overrightarrow{\dim}(\mathbb{1}^I) = 0.$

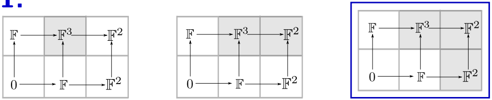
$M =$





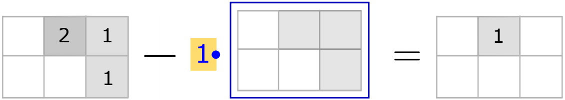
One possible scenario:

Step 1.

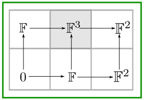


$\mathbf{rk}_M(I) = 2$

Step 2.



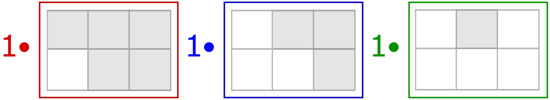
Step 1.



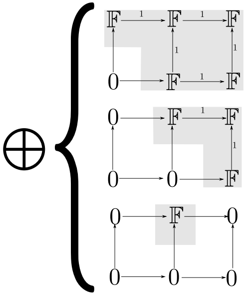
$\mathbf{rk}_M(I) = 3$

Step 2. $\overrightarrow{\dim}(M) - 1 \cdot \overrightarrow{\dim}(\mathbb{1}^I) = 0.$

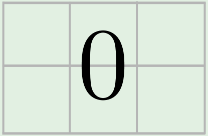
Output:



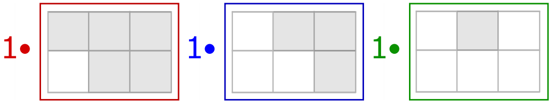
$M =$



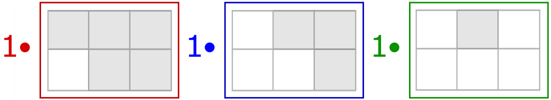
$\overrightarrow{\dim}(M)$



Output:

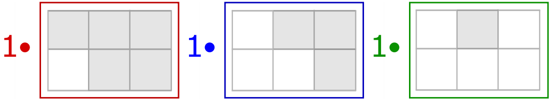


Output:



> **Theorem 2 (Dey, K, Mémoli)**
>
> If M is interval decomposable, then the output is $\mathbf{barc}(M)$.

Output:



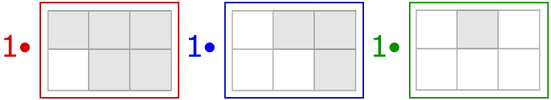
Theorem 2 (Dey, K, Mémoli)

If M is interval decomposable, then the output is $\mathbf{barc}(M)$.

Corollary

If M does not admit the output as its barcode, then M is not interval decomposable.

Output:



Theorem 2 (Dey, K, Mémoli)

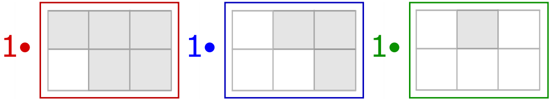
If M is interval decomposable, then the output is $\mathbf{barc}(M)$.

Corollary

If M does not admit the output as its barcode, then M is not interval decomposable.

Algorithm 2. Check whether M is interval decomposable.

Output:



Theorem 2 (Dey, K, Mémoli)

If M is interval decomposable, then the output is $\mathbf{barc}(M)$.

Corollary

If M does not admit the output as its barcode, then M is not interval decomposable.

Algorithm 2. Check whether M is interval decomposable.

Check if each interval I in the output is a summand of M with correct multiplicity, using **[Asashiba et al. 2021]**-algorithm.

Computational complexity

Input: 2-parameter simplicial filtration $\mathcal{F}$.

Let $t := \max\{\#\text{ of points in the indexing poset}, \#\text{ of simplices}\}$.

Computational complexity

Input: 2-parameter simplicial filtration $\mathcal{F}$.

Let $t := \max\{\#$ of points in the indexing poset, $\#$ of simplices$\}$.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time, where $\omega \in [2, 2.373)$, the matrix multiplication exponent.

Computational complexity

Input: 2-parameter simplicial filtration $\mathcal{F}$.

Let $t := \max\{\#\text{ of points in the indexing poset}, \#\text{ of simplices}\}$.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time, where $\omega \in [2, 2.373)$, the matrix multiplication exponent.

It takes $O(t^{2\omega+1})$ using **[Dey, Xin 2022]**'s decomposition algorithm.

Computational complexity

Input: 2-parameter simplicial filtration $\mathcal{F}$.

Let $t := \max\{\#\text{ of points in the indexing poset}, \#\text{ of simplices}\}$.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time, where $\omega \in [2, 2.373)$, the matrix multiplication exponent.

It takes $O(t^{2\omega+1})$ using **[Dey, Xin 2022]**'s decomposition algorithm.

Proposition 2

The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

- **[Asashiba et al. 2021]** requires time exponential in t, since there are exponentially many intervals.

Computational complexity

Input: 2-parameter simplicial filtration $\mathcal{F}$.

Let $t := \max\{\#$ of points in the indexing poset, $\#$ of simplices$\}$.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time, where $\omega \in [2, 2.373)$, the matrix multiplication exponent.

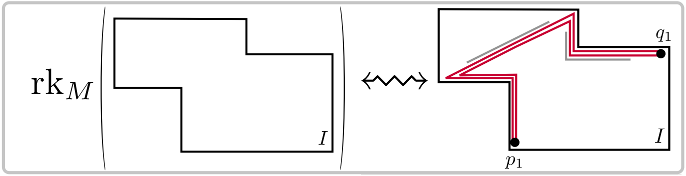
It takes $O(t^{2\omega+1})$ using **[Dey, Xin 2022]**'s decomposition algorithm.

Proposition 2

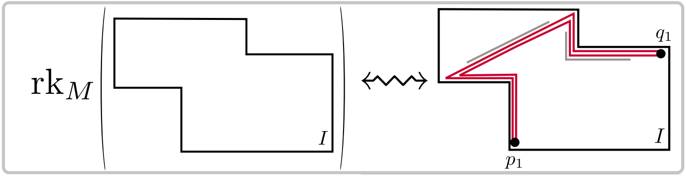
The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

- **[Asashiba et al. 2021]** requires time exponential in t, since there are exponentially many intervals.
- **[Dey, Xin 2022]** requires only $O(t^{2\omega+1})$ time. However, $\mathrm{H}(\mathcal{F})$ must be such that no two generators or relations have the same grade.

Future work

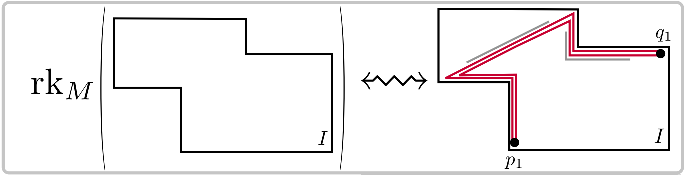


Future work



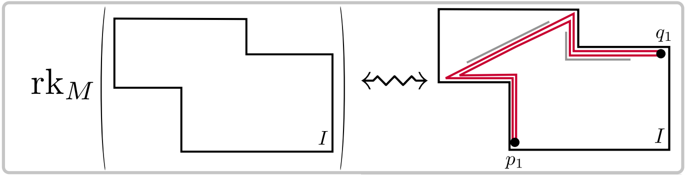
- Beyond 2-parameter persistence. (Ongoing work).

Future work



- Beyond 2-parameter persistence. (Ongoing work).
- Computing $\mathbf{rk}_M : \mathbf{Int}(\mathbb{P}) \to \mathbb{Z}_{\geq 0}$. (Ongoing work).

Future work



- Beyond 2-parameter persistence. (Ongoing work).
- Computing $\mathbf{rk}_M : \mathbf{Int}(\mathbb{P}) \to \mathbb{Z}_{\geq 0}$. (Ongoing work).
- Improvement of the interval decomposability algorithm.

Thank you for paying attention.

Hideto Asashiba, Mickaël Buchet, Emerson G Escolar, Ken Nakashima, and Michio Yoshiwaki.
On interval decomposability of 2d persistence modules.
Computational Geometry, 105:101879, 2022.

Hideto Asashiba, Emerson G Escolar, Ken Nakashima, and Michio Yoshiwaki.
On approximation of 2 d persistence modules by interval-decomposables.
arXiv preprint arXiv:1911.01637, 2019.

Magnus Bakke Botnan, Vadim Lebovici, and Steve Oudot.
On rectangle-decomposable 2-parameter persistence modules.
Discrete & Computational Geometry, pages 1–24, 2022.

Tamal K Dey and Tao Hou.
Fast computation of zigzag persistence.
arXiv preprint arXiv:2204.11080, 2022.

Tamal K Dey, Woojin Kim, and Facundo Mémoli.

Computing generalized rank invariant for 2-parameter persistence modules via zigzag persistence and its applications.
The Proceedings of International Symposium on Computational Geometry, 2022, 2021.

Tamal K Dey and Cheng Xin.
Generalized persistence algorithm for decomposing multi-parameter persistence modules.
arXiv preprint arXiv:1904.03766, 2019.

Woojin Kim and Facundo Mémoli.
Generalized persistence diagrams for persistence modules over posets.
Journal of Applied and Computational Topology, 5(4):533–581, 2021.

Amit Patel.
Generalized persistence diagrams.
Journal of Applied and Computational Topology, pages 1–23, 2018.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time.

1. There are at most $O(t^2)$ interval extension steps.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time.

1. There are at most $O(t^2)$ interval extension steps.
2. For each interval in the extension process, zigzag persistent homology algorithm runs in at most $O(t^{\omega})$ time by [Milosavljevic, Morozov, Skraba 2011].

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time.

1. There are at most $O(t^2)$ interval extension steps.
2. For each interval in the extension process, zigzag persistent homology algorithm runs in at most $O(t^{\omega})$ time by [Milosavljevic, Morozov, Skraba 2011].

Hence, (1)×(2)= $O(t^{\omega+2})$.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time.

① There are at most $O(t^2)$ interval extension steps.

② For each interval in the extension process, zigzag persistent homology algorithm runs in at most $O(t^{\omega})$ time by [Milosavljevic, Morozov, Skraba 2011].

Hence, ①×②$= O(t^{\omega+2})$.

Comparison

$O(t^{2\omega+1})$ using **[Dey, Xin 2022]**-algorithm.

Proposition 1

Given that $\mathrm{H}(\mathcal{F})$ is interval decomposable, the barcode can be computed in $O(t^{\omega+2})$ time.

① There are at most $O(t^2)$ interval extension steps.

② For each interval in the extension process, zigzag persistent homology algorithm runs in at most $O(t^\omega)$ time by [Milosavljevic, Morozov, Skraba 2011].

Hence, ①×②$= O(t^{\omega+2})$.

Comparison

$O(t^{2\omega+1})$ using **[Dey, Xin 2022]**-algorithm.

(Note: **[Dey, Xin 2022]**-algorithm is more general).

Proposition 2

The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

1. Run **Algorithm 1**: $O(t^{\omega+2})$.

Proposition 2

The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

1. Run **Algorithm 1**: $O(t^{\omega+2})$.
2. The output contains $O(t^2)$ intervals.

Proposition 2

The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

1. Run **Algorithm 1**: $O(t^{\omega+2})$.
2. The output contains $O(t^2)$ intervals.
3. For each interval I in the barcode candidate, check whether

$$\underbrace{\mathbb{I}^I \oplus \mathbb{I}^I \oplus \mathbb{I}^I \oplus \cdots \oplus \mathbb{I}^I}_{\mu_I}$$

is the summand of M using results by [Asashiba, Buchet, Escholar, Nakashima, Yoshiwaki 2021]. It takes $O(t^{3\omega})$ in time.

Proposition 2

The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

① Run **Algorithm 1**: $O(t^{\omega+2})$.

② The output contains $O(t^2)$ intervals.

③ For each interval I in the barcode candidate, check whether

$$\underbrace{\mathbb{I}^I \oplus \mathbb{I}^I \oplus \mathbb{I}^I \oplus \cdots \oplus \mathbb{I}^I}_{\mu_I}$$

is the summand of M using results by [Asashiba, Buchet, Escholar, Nakashima, Yoshiwaki 2021]. It takes $O(t^{3\omega})$ in time.

In sum, ① + (② × ③) $= O(t^{3\omega+2})$.

Proposition 2

The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

1. Run **Algorithm 1**: $O(t^{\omega+2})$.
2. The output contains $O(t^2)$ intervals.
3. For each interval I in the barcode candidate, check whether

$$\underbrace{\mathbb{I}^I \oplus \mathbb{I}^I \oplus \mathbb{I}^I \oplus \cdots \oplus \mathbb{I}^I}_{\mu_I}$$

   is the summand of M using results by [Asashiba, Buchet, Escholar, Nakashima, Yoshiwaki 2021]. It takes $O(t^{3\omega})$ in time.

In sum, ① + (② × ③) $= O(t^{3\omega+2})$.

Comparison

- **[Asashiba et al. 2021]** requires time exponential in t, since there are exponentially many intervals.

Proposition 2

The interval decomposability of $\mathrm{H}(\mathcal{F})$ can be determined in $O(t^{3\omega+2})$.

① Run **Algorithm 1**: $O(t^{\omega+2})$.

② The output contains $O(t^2)$ intervals.

③ For each interval I in the barcode candidate, check whether

$$\underbrace{\mathbb{I}^I \oplus \mathbb{I}^I \oplus \mathbb{I}^I \oplus \cdots \oplus \mathbb{I}^I}_{\mu_I}$$

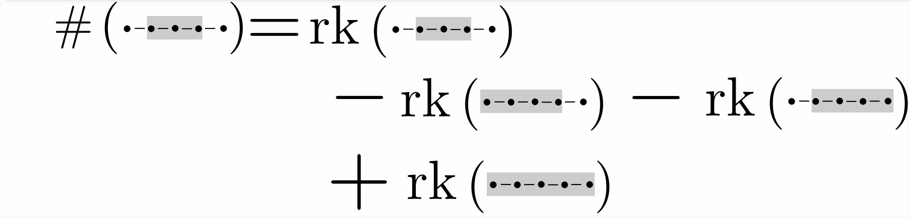
is the summand of M using results by [Asashiba, Buchet, Escholar, Nakashima, Yoshiwaki 2021]. It takes $O(t^{3\omega})$ in time.

In sum, ①+(②×③) $= O(t^{3\omega+2})$.

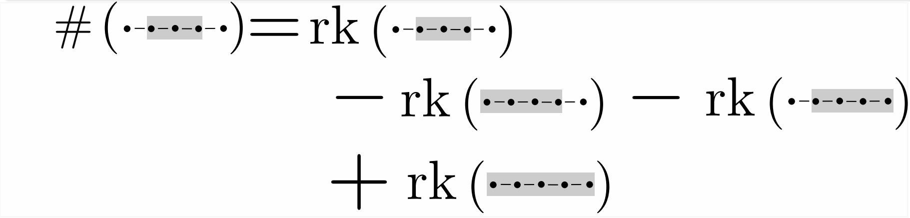
Comparison

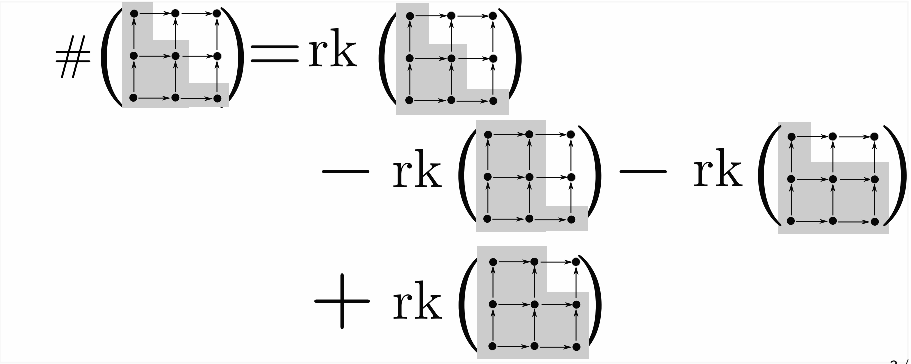
- **[Asashiba et al. 2021]** requires time exponential in t, since there are exponentially many intervals.
- **[Dey, Xin 2019]** requires only $O(t^{2\omega+1})$ time. However, $\mathrm{H}(\mathcal{K})$ must be such that no two generators or relations have the same grade.

How to compute the barcode?

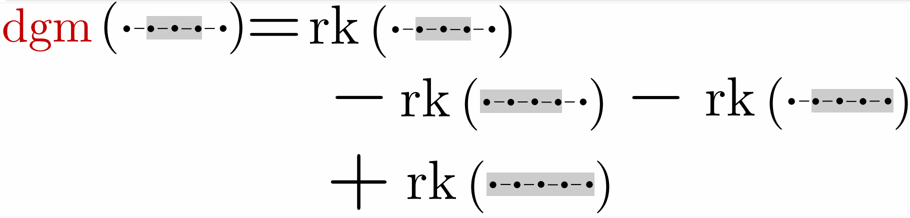

#
=
rk
− rk
− rk
+ rk

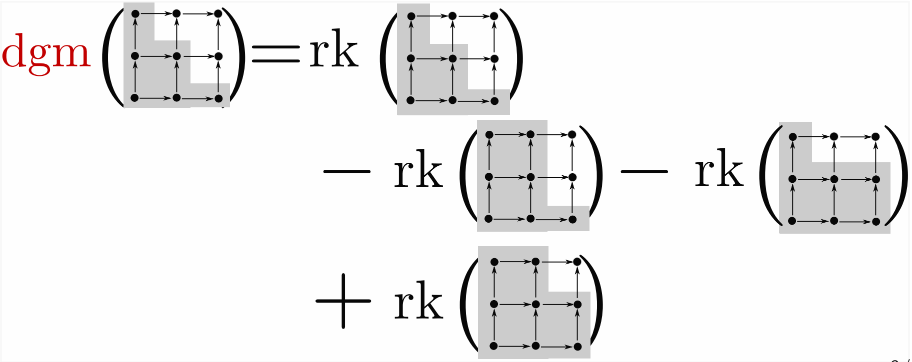
How to compute the barcode?



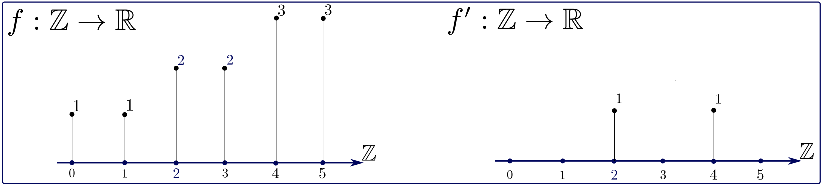


How to define the generalized persistence diagram?

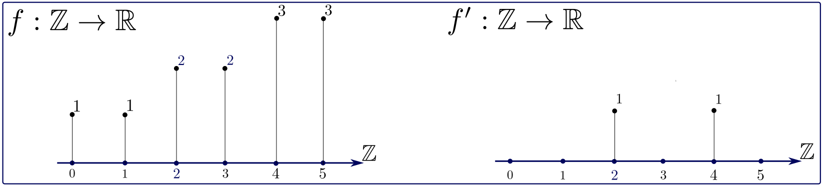




Möbius Inversion: discrete derivative

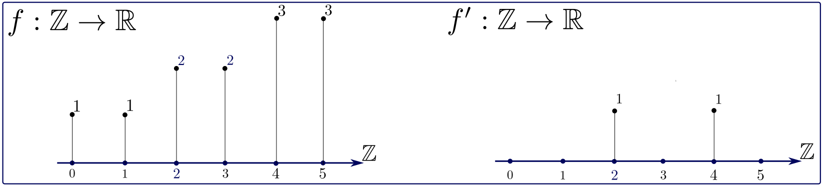


Möbius Inversion: discrete derivative



Poset	A given function	Möbius inversion
$(\mathbb{Z}, \leq)$	$f : \mathbb{Z} \to \mathbb{R}$	$f'(n) = f(n) - f(n-1)$

Möbius Inversion: discrete derivative



Poset	A given function	Möbius inversion
$(\mathbb{Z}, \leq)$	$f : \mathbb{Z} \to \mathbb{R}$	$f'(n) = f(n) - f(n-1)$
$(\mathbf{Int}(\mathbb{P}), \supseteq)$	$\mathbf{rk}_M : \mathbf{Int}(\mathbb{P}) \to \mathbb{Z}$	$\mathbf{dgm}_M : \mathbf{Int}(\mathbb{P}) \to \mathbb{Z}$