PERSISTENT RELATIVE HOMOLOGY FOR TOPOLOGICAL DATA ANALYSIS

Christian Lentz

April 17, 2024

BACKGROUND

TOPOLOGY

Topology is concerned with certain *qualitative properties* of spaces/objects that are invariant (do not change) under certain types of *continuous transformations* (functions).

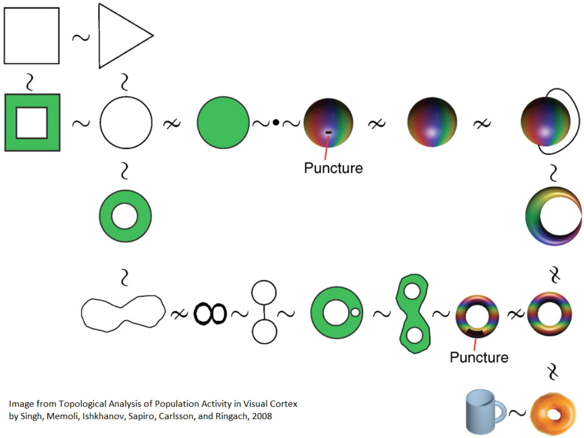


Image from Topological Analysis of Population Activity in Visual Cortex
by Singh, Memoli, Ishkhanov, Sapiro, Carlsson, and Ringach, 2008

SIMPLICIAL COMPLEXES

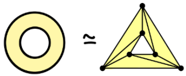
- Study a complicated structure by breaking it into "simple pieces" called **simplices**.

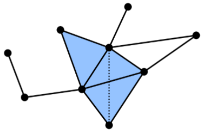
Definition

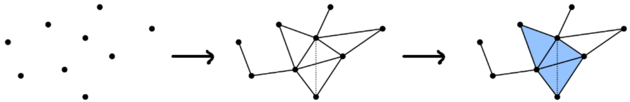
A **simplicial complex** is a subspace $K \subseteq \mathbb{R}^n$ such that

1. if $\sigma \in K$ and $\tau \subset \sigma$ then $\tau \in K$.
2. if $\sigma, \tau \in K$ then $\sigma \cap \tau$ is empty or a subsimplex of both.

- Graphs/networks, topological spaces, point cloud data, etc.



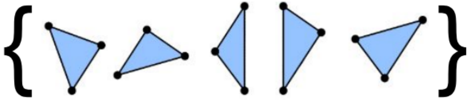




Definition

Define $C_n(K)$ to be the $\mathbb{Z}_2$ vector space whose basis is the set of n-simplices in K.

- A linear combination of n-simplices is an n-**chain**.
- For example, $C_2(K)$ is spanned by the following basis of 2-simplices:



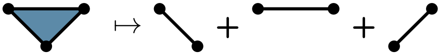
BOUNDARY OPERATORS

Definition

Define the (alternating) **boundary operator** as a linear transformation $\partial_n : C_n(K) \to C_{n-1}(K)$ given by

$$[v_0 v_1 \dots v_n] \mapsto \sum_{j=0}^{n} (-1)^j [v_0 v_1 \dots \hat{v}_j \dots v_n].$$

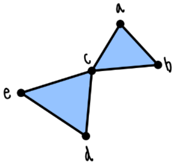
- Map an n-simplex to an $(n-1)$-chain which is its boundary.



- Every function has a
  - **Kernel**: All inputs that map to zero.
  - **Image**: All outputs.
- Elements in the kernel of a boundary operator are called **cycles**, and elements in the image are **boundaries**.

BOUNDARY OPERATORS (EXAMPLE)

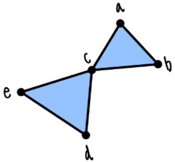
What is ∂_1 for the following simplicial complex K?



- Recall $\partial_1 : C_1(K) \to C_0(K)$
- $C_1(K) = \{[ab], [ac], [bc], [cd], [ce], [de]\}$
- $C_0(K) = \{a, b, c, d, e\}$

BOUNDARY OPERATORS (EXAMPLE)

What is ∂_1 for the following simplicial complex K?

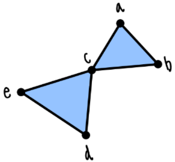


- We have $\partial_1([v_0 v_1]) = [\hat{v}_0 v_1] - [v_0 \hat{v}_1]$, now fill out each column of the matrix!

$$\partial_1 = \begin{array}{c c} & \begin{array}{cccccc} [ab] & [ac] & [bc] & [cd] & [ce] & [de] \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} & & & & & \\ \\ \\ \\ \\ \end{pmatrix} \end{array}$$

BOUNDARY OPERATORS (EXAMPLE)

What is ∂_1 for the following simplicial complex K?

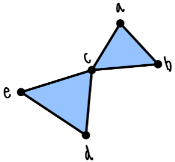


- We have $\partial_1([v_0 v_1]) = [\hat{v}_0 v_1] - [v_0 \hat{v}_1]$, now fill out each column of the matrix!

$$\partial_1 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{cccccc} [ab] & [ac] & [bc] & [cd] & [ce] & [de] \end{array} \\ \begin{pmatrix} -1 & & & & & \\ 1 & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix} \end{array}$$

BOUNDARY OPERATORS (EXAMPLE)

What is ∂_1 for the following simplicial complex K?

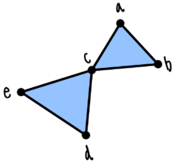


- We have $\partial_1([v_0 v_1]) = [\hat{v}_0 v_1] - [v_0 \hat{v}_1]$, now fill out each column of the matrix!

$$\partial_1 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{cccccc} [ab] & [ac] & [bc] & [cd] & [ce] & [de] \end{array} \\ \begin{pmatrix} -1 & -1 & & & & \\ 1 & & & & & \\ & 1 & & & & \\ & & & & & \\ & & & & & \end{pmatrix} \end{array}$$

BOUNDARY OPERATORS (EXAMPLE)

What is ∂_1 for the following simplicial complex K?

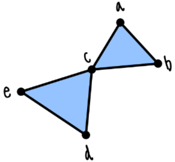


- We have $\partial_1([v_0 v_1]) = [\hat{v}_0 v_1] - [v_0 \hat{v}_1]$, now fill out each column of the matrix!

$$
\partial_1 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c}
\begin{array}{cccccc} [ab] & [ac] & [bc] & [cd] & [ce] & [de] \end{array} \\
\begin{pmatrix}
-1 & -1 & & & & \\
1 & & -1 & & & \\
 & 1 & 1 & & & \\
 & & & & & \\
 & & & & &
\end{pmatrix}
\end{array}
$$

BOUNDARY OPERATORS (EXAMPLE)

What is ∂_1 for the following simplicial complex K?

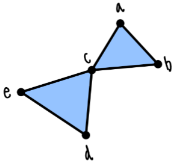


- We have $\partial_1([v_0 v_1]) = [\hat{v}_0 v_1] - [v_0 \hat{v}_1]$, now fill out each column of the matrix!

$$\partial_1 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{cccccc} [ab] & [ac] & [bc] & [cd] & [ce] & [de] \end{array} \\ \begin{pmatrix} -1 & -1 & & & & \\ 1 & & -1 & & & \\ & 1 & 1 & -1 & -1 & \\ & & & 1 & & \\ & & & & 1 & \end{pmatrix} \end{array}$$

BOUNDARY OPERATORS (EXAMPLE)

What is ∂_1 for the following simplicial complex K?

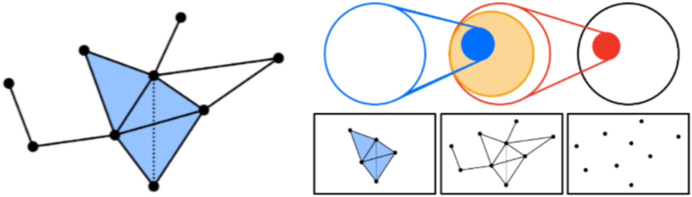


- We have $\partial_1([v_0 v_1]) = [\hat{v}_0 v_1] - [v_0 \hat{v}_1]$, now fill out each column of the matrix!

$$\partial_1 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{cccccc} [ab] & [ac] & [bc] & [cd] & [ce] & [de] \end{array} \\ \begin{pmatrix} -1 & -1 & & & & \\ 1 & & -1 & & & \\ & 1 & 1 & -1 & -1 & \\ & & & 1 & & -1 \\ & & & & 1 & 1 \end{pmatrix} \end{array}$$

CHAIN COMPLEXES

- A **chain complex** is a sequence of boundary operators where $\partial_{n-1}\partial_n = 0$, which means *a boundary has no boundary*.



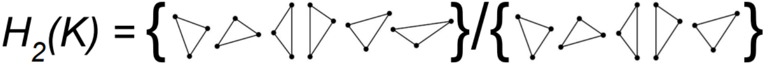
$$\ldots C_2(K) \xrightarrow{\partial_2} C_1(K) \xrightarrow{\partial_1} C_0(K) \ldots$$

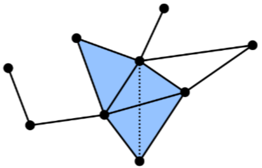
- Notice: $\mathrm{Im}(\partial_{n+1}) \subseteq \mathrm{Ker}(\partial_n)$
- Every $(n+1)$-boundary is also an n-cycle, but the converse is not always true.

HOMOLOGY GROUPS

- A **homology group** $H_n(K)$ describes all of the n-dimensional holes in the simplicial complex K.
- Deterrmine which cycles are *not* boundaries.

$$H_n(K) = \mathrm{Ker}(\partial_{n-1})/\mathrm{Im}(\partial_n)$$
$$= \mathit{cycles} - \mathit{boundaries}$$

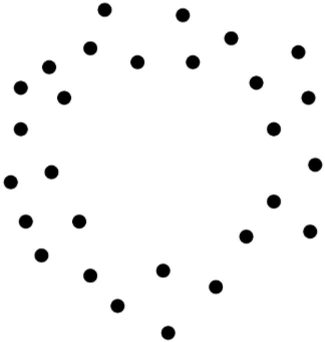




PERSISTENT HOMOLOGY

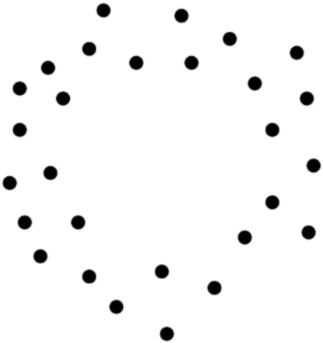
WHAT IS PERSISTENT HOMOLOGY?

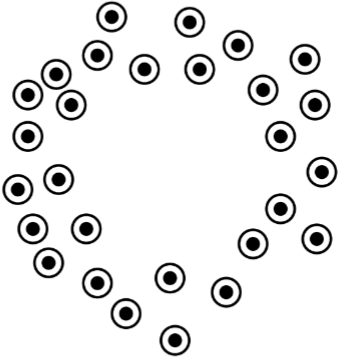
- Persistent homology (PH) is a tool in topological data analysis (TDA) used to study the shape of data.
- Apply homology to a sequence of nested topological spaces called a **filtered topological space**.
- Features which are *born* and *die* quickly are **noise**.
- Associate features of interest as topological holes.
- Important features persist throughout filtration.



Let ϵ_1 be a positive real number. Place a disk with a radius of ϵ_1 around each point.

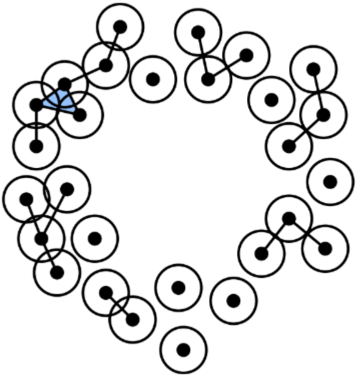
Let ϵ_1 be a positive real number. Place a disk with a radius of ϵ_1 around each point.

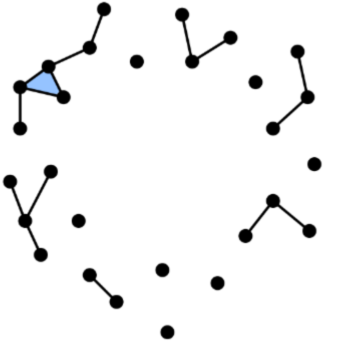




This is a simplicial complex. Call it K_1.

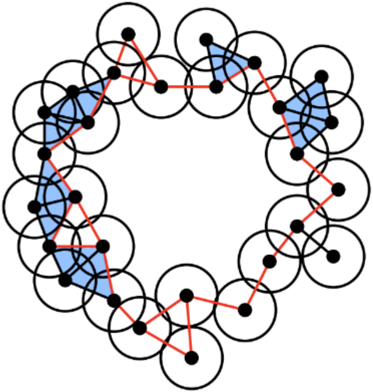
Now let $\epsilon_1 \leq \epsilon_2$ and repeat, with the rule that anytime $k+1$ points are pairwise within $2\epsilon_2$ of each other, form a k simplex:

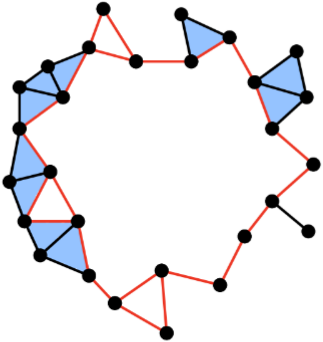




Notice that $K_1 \subseteq K_2$. Also, there are now **connected components** in K_2.

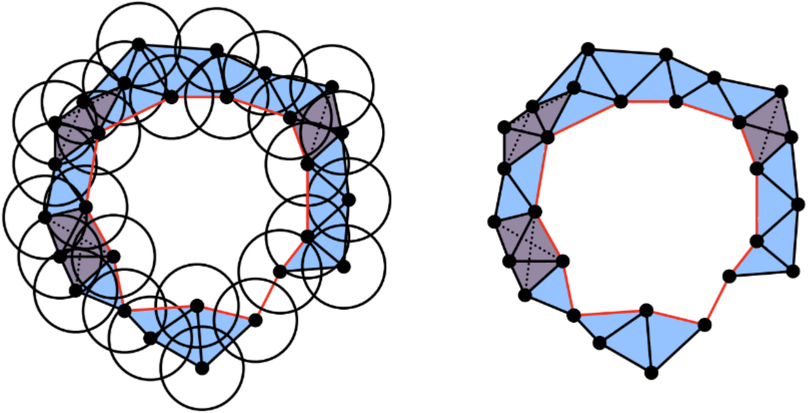
Now K_{ϵ_3}:



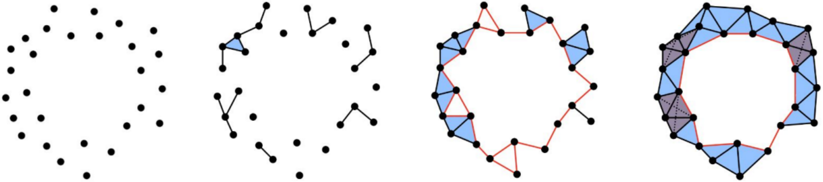


Notice that K_3 has one-dimensional features, called **loops**.

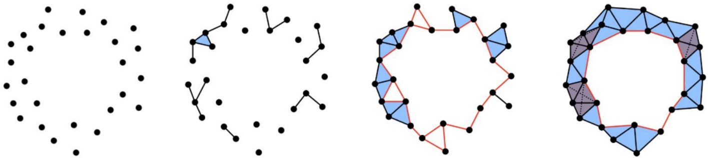
Finally, we have simplicial complex K_4:



The **scale parameter** $\epsilon_1 \leq \epsilon_2 \leq \epsilon_3 \leq \epsilon_4$ gives us the following filtration.



Notice that homological features (connected components, loops, etc) emerge. Some of them stay, and some of them are removed quickly! This method is called a **Čech complex**.



Definition

A **finite filtration** on a simplicial complex K, denoted $F_\bullet K$, is given by $F_1 K \subseteq F_2 K \subseteq \cdots \subseteq F_N K$, where $F_N K = K$.

Definition

A simplicial complex K equipped with a filtration F is called a **filtered simplicial complex**.

Definition

Say that $\sigma \in K$ born at $F_t K$ has a **filtration value** $b(\sigma) = t$. Thus, $F_t K = \{\sigma \in K : b(\sigma) \leq t\}$.

PERSISTENT RELATIVE HOMOLOGY

QUOTIENT SPACES

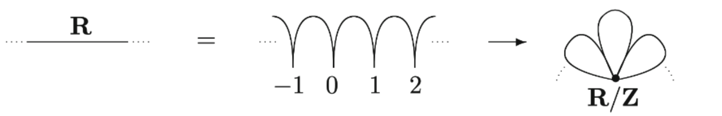
Definition

Suppose topological spaces X and A such that $A \subseteq X$. Then the **quotient space** is defined as

$$X/A = (X \setminus A) \sqcup *$$

where * is a single point.

Example (The Infinite Bouquet[1])



[1] Image from *Essential Topology* by Martin D. Crossley (2005).

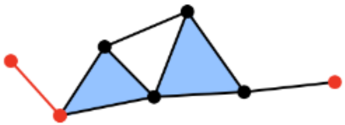
RELATIVE CHAINS

- Suppose two simplicial complexes K and K_0 where $K_0 \subset K$.

Definition

The **relative chain vector space** is the quotient vector space $C_n(K, K_0) = C_n(K)/C_n(K_0)$, which describes the span of all **relative n-chains** in K/K_0.

- $C_n(K, K_0)$ partitions a basis for $C_n(K)$ into cosets (or equivalence classes) of the form $c + C_n(K_0)$.
- Equivalence classes $\{[c]\}$ give a basis for all chains in $K - K_0$.



RELATIVE HOMOLOGY

- **Relative Homology** is the homology of K/K_0.
- Cycles and boundaries look different in this setting.

Definition

A *Relative n-Cycle* is any n-chain $\alpha \in C_n(K)$ such that $\partial_n(\alpha) \in C_{n-1}(K_0)$. In words, any n-chain with a boundary in the subspace K_0.

RELATIVE HOMOLOGY

- **Relative Homology** is the homology of K/K_0.
- Cycles and boundaries look different in this setting.

Definition

A *Relative n-Cycle* is any n-chain $\alpha \in C_n(K)$ such that $\partial_n(\alpha) \in C_{n-1}(K_0)$. In words, any n-chain with a boundary in the subspace K_0.

Definition

A *Relative n-Boundary* is any relative n-cycle $\alpha = \partial_{n+1}(\beta) + \gamma$ for some $\beta \in C_{n+1}(K)$ and $\gamma \in C_n(K_0)$. In words, any n-cycle which differs from an *absolute boundary* by a chain in the subspace K_0.

RELATIVE HOMOLOGY

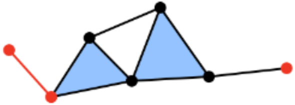
- **Relative Homology** is the homology of K/K_0.
- Cycles and boundaries look different in this setting.

Definition

A *Relative n-Cycle* is any n-chain $\alpha \in C_n(K)$ such that $\partial_n(\alpha) \in C_{n-1}(K_0)$. In words, any n-chain with a boundary in the subspace K_0.

Definition

A *Relative n-Boundary* is any relative n-cycle $\alpha = \partial_{n+1}(\beta) + \gamma$ for some $\beta \in C_{n+1}(K)$ and $\gamma \in C_n(K_0)$. In words, any n-cycle which differs from an *absolute boundary* by a chain in the subspace K_0.



RELATIVE HOMOLOGY

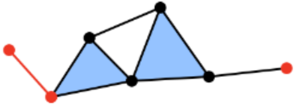
- **Relative Homology** is the homology of K/K_0.
- Cycles and boundaries look different in this setting.

Definition

A *Relative n-Cycle* is any n-chain $\alpha \in C_n(K)$ such that $\partial_n(\alpha) \in C_{n-1}(K_0)$. In words, any n-chain with a boundary in the subspace K_0.

Definition

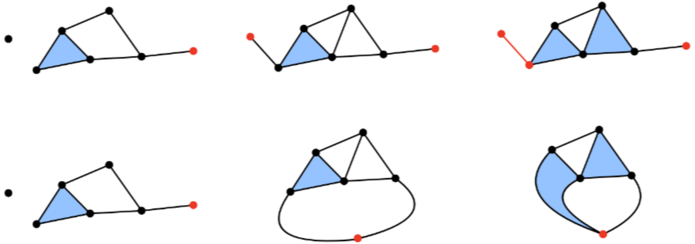
A *Relative n-Boundary* is any relative n-cycle $\alpha = \partial_{n+1}(\beta) + \gamma$ for some $\beta \in C_{n+1}(K)$ and $\gamma \in C_n(K_0)$. In words, any n-cycle which differs from an *absolute boundary* by a chain in the subspace K_0.



$$\partial_1(\alpha) = \partial_1(\partial_2(\beta)) + \partial_1(\gamma) = \partial_1(\gamma) \in C_0(K_0).$$

PERSISTENT RELATIVE HOMOLOGY

- Given $F_\bullet K$ and $G_\bullet K_0$.
- **Persistent Relative Homology** (PRH) is the homology of a filtered quotient space K/K_0.
- Require that $G_t K_0 \subseteq F_t K$ for each time-step t.
- Do not require that $\sigma \in K$ satisfy $b_F(\sigma) = b_G(\sigma)$.



THE U-MATCH DECOMPOSITION

DEFINITION

- Assume D is the **block boundary matrix** of a chain complex, so D is square and $D^2 = 0$.

$$D = \begin{pmatrix} 0 & \partial_1 & & & \\ & 0 & \partial_2 & & \\ & & \ddots & \ddots & \\ & & & 0 & \partial_N \\ & & & & 0 \end{pmatrix}$$

DEFINITION

- Assume D is the **block boundary matrix** of a chain complex, so D is square and $D^2 = 0$.

$$D = \begin{pmatrix} 0 & \partial_1 & & & \\ & 0 & \partial_2 & & \\ & & \ddots & \ddots & \\ & & & 0 & \partial_N \\ & & & & 0 \end{pmatrix}$$

- Reduce D bottom to top and left to right. $\mathcal{T}^{-1}$ records row operations, and $\mathcal{S}$ records column operations.

$$\begin{pmatrix} D & I_n \\ I_m & 0 \end{pmatrix} \mapsto \begin{pmatrix} M & \mathcal{T}^{-1} \\ \mathcal{S} & 0 \end{pmatrix}$$

DEFINITION

- Assume D is the **block boundary matrix** of a chain complex, so D is square and $D^2 = 0$.

$$D = \begin{pmatrix} 0 & \partial_1 & & & \\ & 0 & \partial_2 & & \\ & & \ddots & \ddots & \\ & & & 0 & \partial_N \\ & & & & 0 \end{pmatrix}$$

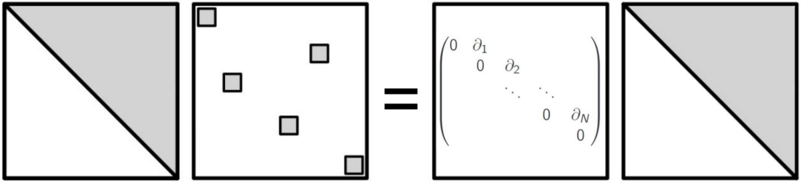
- Reduce D bottom to top and left to right. $\mathcal{T}^{-1}$ records row operations, and $\mathcal{S}$ records column operations.

$$\begin{pmatrix} D & I_n \\ I_m & 0 \end{pmatrix} \mapsto \begin{pmatrix} M & \mathcal{T}^{-1} \\ \mathcal{S} & 0 \end{pmatrix}$$

- A **U-Match Decomposition** is a tuple of matrices $(\mathcal{T}, M, D, \mathcal{S})$ which satisfy the following three conditions:
  - $\mathcal{T}M = D\mathcal{S}$
  - M is a **matching matrix**
  - $\mathcal{T}$ and $\mathcal{S}$ are both upper triangular and invertible

U-MATCH DECOMPOSITION

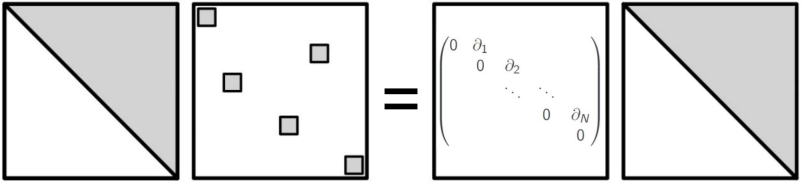
- Persistence algorithms use matrix decomposition techniques.
- $\mathcal{T}$ and $\mathcal{S}$ contain information about homology.



$$\mathcal{T} \times M = D \times \mathcal{S}$$

U-MATCH DECOMPOSITION

- Persistence algorithms use matrix decomposition techniques.
- $\mathcal{T}$ and $\mathcal{S}$ contain information about homology.



$$\mathcal{T} \times M = D \times \mathcal{S}$$

- Ordering the rows and columns of D carefully allows us to compute persistent homology.

U-MATCH PROPERTIES

- Let $\mathcal{T}M = D\mathcal{S}$ be a U-match decomposition, where D is the block boundary matrix of a chain complex.

U-MATCH PROPERTIES

- Let $\mathcal{T}M = D\mathcal{S}$ be a U-match decomposition, where D is the block boundary matrix of a chain complex. Let $r_\bullet$ and $c_\bullet$ denote, respectively, the set of indices of **nonzero rows** and **columns** of the matching matrix M.

U-MATCH PROPERTIES

- Let $\mathcal{T}M = D\mathcal{S}$ be a U-match decomposition, where D is the block boundary matrix of a chain complex. Let $r_\bullet$ and $c_\bullet$ denote, respectively, the set of indices of **nonzero rows** and **columns** of the matching matrix M.

Lemma

The set of indices $r_\bullet$ and $c_\bullet$ are disjoint. Hence, $r_\bullet \subseteq \overline{c_\bullet}$.

Outline of proof.

- $\mathcal{T}M = D\mathcal{S} \Rightarrow \mathcal{S}^{-1}\mathcal{T}M = \mathcal{S}^{-1}D\mathcal{S}$.
- $(\mathcal{S}^{-1}D\mathcal{S})^2 = \mathcal{S}^{-1}D^2\mathcal{S} = 0$.
- $(\mathcal{S}^{-1}\mathcal{T}M)^2 = 0$ implies that indices of nonzero rows and columns of $\mathcal{S}^{-1}\mathcal{T}M$ are disjoint.

- Let $\mathcal{T}M = D\mathcal{S}$ be a U-match decomposition, where D is the block boundary matrix of a chain complex. Let $r_\bullet$ and $c_\bullet$ denote, respectively, the set of indices of **nonzero rows** and **columns** of the matching matrix M.

Corollary

Columns of $\mathcal{T}$ indexed by the set $r_\bullet$ give a basis for $\mathrm{Im}(D)$, which are the boundaries.

Outline of Proof:

- $COL_j(\mathcal{T}M) = COL_j(D\mathcal{S}) \Rightarrow COL_j(\mathcal{T}M) = D \cdot COL_j(\mathcal{S})$. So $COL_j(\mathcal{T}M)$ is the boundary of some column of S.
- Can write $COL_j(\mathcal{T}M) = COL_i(\mathcal{T}) \cdot M$ where i corresponds to nonzero row in M.

- Let $\mathcal{T}M = D\mathcal{S}$ be a U-match decomposition, where D is the block boundary matrix of a chain complex. Let $r_\bullet$ and $c_\bullet$ denote, respectively, the set of indices of **nonzero rows** and **columns** of the matching matrix M.

Corollary

Columns of $\mathcal{S}$ indexed by the set $\overline{c_\bullet}$ contain a basis for $\mathrm{Ker}(D)$, which are the cycles.

Outline of Proof:

- Assume $j \in \overline{c_\bullet}$.
- $D \cdot COL_j(\mathcal{S}) = \mathcal{T} \cdot COL_j(M) = \mathcal{T} \cdot \vec{0} = \vec{0}$.

U-MATCH PROPERTIES (CONTINUED)

- U-Match allows us to compute **matched bases** for cycles and boundaries.
- This means a set of basis vectors for $\mathrm{Im}(D)$ is a subset of a set of basis vectors for $\mathrm{Ker}(D)$.
- How? Prove this by construction!
  - Construct a matrix J from the matrix $\mathcal{S}$ with the substitution

$$COL_{r_j}(\mathcal{S}) \mapsto COL_{c_j}(\mathcal{T}M).$$

  - Columns of J contain a basis for both $\mathrm{Im}(D)$ and $\mathrm{Ker}(D)$.
    - $COL_{\overline{c_\bullet}}(J) = \mathrm{Ker}(D)$
    - $COL_{r_\bullet}(J) = \mathrm{Im}(D)$
  - Recall that $r_\bullet \subseteq \overline{c_\bullet}$.

U-MATCH FOR PERSISTENCE

- Suppose a filtered simplicial complex $F_\bullet K$.
- Construct the block boundary matrix where filtration value increases with row and column indices.
- This ordering is carried over to $\mathcal{T}$, $\mathcal{S}$ and M:

$$\begin{array}{cc} & c \in F_\bullet K \\ s \in F_\bullet K & \begin{pmatrix} \quad & \mathcal{T} & \quad \end{pmatrix} \end{array} \qquad \begin{array}{cc} & c \in F_\bullet K \\ c \in F_\bullet K & \begin{pmatrix} \quad & M & \quad \end{pmatrix} \end{array}$$

$$\begin{array}{cc} & s \in F_\bullet K \\ s \in F_\bullet K & \begin{pmatrix} \quad & D & \quad \end{pmatrix} \end{array} \qquad \begin{array}{cc} & c \in F_\bullet K \\ s \in F_\bullet K & \begin{pmatrix} \quad & \mathcal{S} & \quad \end{pmatrix} \end{array}$$

THE U-MATCH PRH ALGORITHM

A high-level overview:

A high-level overview:

1. Permute rows of block boundary matrix D to obtain a **relative boundary matrix** $\mathcal{D}$.

THE ALGORITHM

A high-level overview:

1. Permute rows of block boundary matrix D to obtain a **relative boundary matrix** $\mathcal{D}$.
2. Perform a U-Match on $\mathcal{D}$ to get, $\mathcal{T}M = \mathcal{D}\mathcal{S}$.

A high-level overview:

1. Permute rows of block boundary matrix D to obtain a **relative boundary matrix** $\mathcal{D}$.
2. Perform a U-Match on $\mathcal{D}$ to get, $\mathcal{T}M = \mathcal{D}\mathcal{S}$.
3. Permute columns of $\mathcal{T}$ to obtain a matrix $\mathcal{A}$, and columns of $\mathcal{S}$ to obtain a matrix $\mathcal{B}$.

THE ALGORITHM

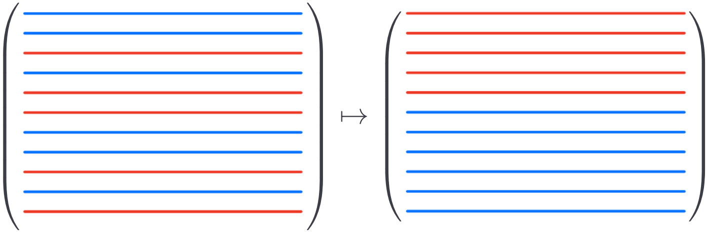
A high-level overview:

1. Permute rows of block boundary matrix D to obtain a **relative boundary matrix** $\mathcal{D}$.
2. Perform a U-Match on $\mathcal{D}$ to get, $\mathcal{T}M = \mathcal{D}\mathcal{S}$.
3. Permute columns of $\mathcal{T}$ to obtain a matrix $\mathcal{A}$, and columns of $\mathcal{S}$ to obtain a matrix $\mathcal{B}$.
4. Perform a U-Match on $\mathcal{A}^{-1}\mathcal{B}$ to obtain $\mathscr{T}\mathcal{M} = (\mathcal{A}^{-1}\mathcal{B})\mathscr{S}$.

Result: One single matrix whose columns contain a filtered basis for the relative cycles and relative boundaries!

STEP 1: THE BOUNDARY MATRIX

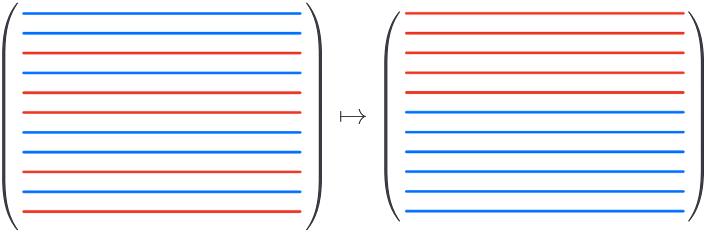
- Suppose you have the boundary matrix of a filtered simplicial complex $F_\bullet K$, and you also have a filtered subcomplex $G_\bullet K_0$.
- Permute rows (top to bottom) to **respect** birth of simplices in $G_\bullet K_0$.



- Why?

STEP 1: THE BOUNDARY MATRIX

- Suppose you have the boundary matrix of a filtered simplicial complex $F_\bullet K$, and you also have a filtered subcomplex $G_\bullet K_0$.
- Permute rows (top to bottom) to **respect** birth of simplices in $G_\bullet K_0$.



- Why? Relative chains correspond to cosets $c + C_n(K_0)$, and U-Match reduces rows from bottom to top!

STEP 2: U-MATCH

- The U-Match $\mathcal{T}M = \mathcal{D}\mathcal{S}$ has a few key differences since we use $\mathcal{D}$ rather than D.

$$\begin{array}{cc} & c \in G_\bullet K_0 \\ s \in G_\bullet K_0 & \left(\begin{array}{ccc} \quad & \mathcal{T} & \quad \end{array}\right) \end{array} \qquad \begin{array}{cc} & c \in F_\bullet K \\ c \in G_\bullet K_0 & \left(\begin{array}{ccc} \quad & M & \quad \end{array}\right) \end{array}$$

$$\begin{array}{cc} & s \in F_\bullet K \\ s \in G_\bullet K_0 & \left(\begin{array}{ccc} \quad & \mathcal{D} & \quad \end{array}\right) \end{array} \qquad \begin{array}{cc} & c \in F_\bullet K \\ s \in F_\bullet K & \left(\begin{array}{ccc} \quad & \mathcal{S} & \quad \end{array}\right) \end{array}$$

- Using this modified U-Match, we can prove the following:

Proposition

Suppose that $\mathcal{V}_{K_0}$ is a vector space with dimension i spanning all chains in K_0. Then the first i columns of $\mathcal{T}$ from the U-Match $\mathcal{T}M = \mathcal{D}\mathcal{S}$ are a basis which spans $\mathcal{V}_{K_0}$.

Lemma

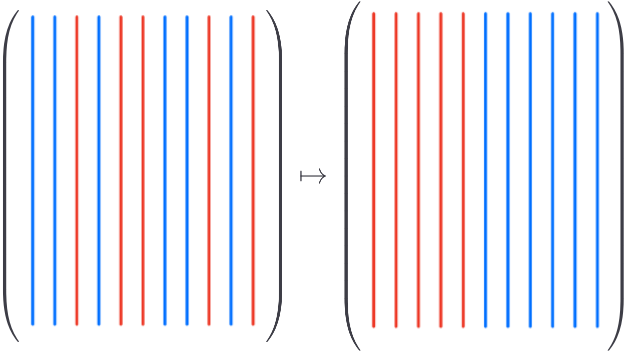
Suppose a U-Match $\mathcal{T}M = \mathcal{D}\mathcal{S}$. Then

(a) the columns of $\mathcal{S}$ contain a basis for the relative cycles denoted $\mathrm{RelKer}(\mathcal{D})$.

(b) the columns of $\mathcal{T}$ contain a basis for the relative boundaries denoted $\mathrm{RelIm}(\mathcal{D})$.

STEP 3: PERMUTE COLUMNS

- The previous results show that this method can compute relative homology. How do we turn this into persistent relative homology?
- Permute columns of $\mathcal{T}$ and $\mathcal{S}$ to **respect** the birth of relative features. Call these $\mathcal{A}$ and $\mathcal{B}$ respectively.



ONE MORE U-MATCH PROPERTY

- A is a square, invertible matrix of size $m \times m$.
- B is a (not necessarily square) matrix of size $m \times n$.
- $F_\bullet$ is a filtration on a vector space $\mathbb{K}^m$ such that $F_i\mathbb{K}^m$ describes the span of the first i columns of A.
- Similarly, define $G_\bullet$ to be a filtration on the columns of B.
- If the columns of B do not span the columns of A, let $G_{n+1} = \mathbb{K}^m$ to ensure $G_\bullet$ terminates.

ONE MORE U-MATCH PROPERTY

- A is a square, invertible matrix of size $m \times m$.
- B is a (not necessarily square) matrix of size $m \times n$.
- $F_\bullet$ is a filtration on a vector space $\mathbb{K}^m$ such that $F_i\mathbb{K}^m$ describes the span of the first i columns of A.
- Similarly, define $G_\bullet$ to be a filtration on the columns of B.
- If the columns of B do not span the columns of A, let $G_{n+1} = \mathbb{K}^m$ to ensure $G_\bullet$ terminates.

Lemma (Basis Matching)

Assume the above conditions hold. It follows that, given the U-Match $\mathcal{T}M = (A^{-1}B)\mathcal{S}$, then the columns of $A\mathcal{T}$ contain a basis for each F_i and G_j for $i, j \in \{1, ..., m\}$.

Theorem

Let K and K_0 be simplicial complexes equipped with finite filtrations $F_\bullet K$ and $G_\bullet K_0$, and suppose that for any filtration value t we have $G_t K_0 \subseteq F_t K$. Apply the following steps:

STEP 4: U-MATCH

Theorem

Let K and K_0 be simplicial complexes equipped with finite filtrations $F_\bullet K$ and $G_\bullet K_0$, and suppose that for any filtration value t we have $G_t K_0 \subseteq F_t K$. Apply the following steps:

1. Construct relative boundary matrix $\mathcal{D}$.

STEP 4: U-MATCH

Theorem

Let K and K_0 be simplicial complexes equipped with finite filtrations $F_\bullet K$ and $G_\bullet K_0$, and suppose that for any filtration value t we have $G_t K_0 \subseteq F_t K$. Apply the following steps:

1. Construct relative boundary matrix $\mathcal{D}$.
2. Perform a U-Match decomposition $\mathcal{T}M = \mathcal{D}\mathcal{S}$.

Theorem

Let K and K_0 be simplicial complexes equipped with finite filtrations $F_\bullet K$ and $G_\bullet K_0$, and suppose that for any filtration value t we have $G_t K_0 \subseteq F_t K$. Apply the following steps:

1. Construct relative boundary matrix $\mathcal{D}$.
2. Perform a U-Match decomposition $\mathcal{T}M = \mathcal{D}\mathcal{S}$.
3. Obtain $\mathcal{A}$ and $\mathcal{B}$ by permuting columns of $\mathcal{T}$ and $\mathcal{S}$.

STEP 4: U-MATCH

Theorem

Let K and K_0 be simplicial complexes equipped with finite filtrations $F_\bullet K$ and $G_\bullet K_0$, and suppose that for any filtration value t we have $G_t K_0 \subseteq F_t K$. Apply the following steps:

1. Construct relative boundary matrix $\mathcal{D}$.
2. Perform a U-Match decomposition $\mathcal{T} M = \mathcal{D} \mathcal{S}$.
3. Obtain $\mathcal{A}$ and $\mathcal{B}$ by permuting columns of $\mathcal{T}$ and $\mathcal{S}$.
4. Perform a U-Match Decomposition $\mathscr{T} \mathcal{M} = (\mathcal{A}^{-1}\mathcal{B})\mathscr{S}$.

Theorem

Let K and K_0 be simplicial complexes equipped with finite filtrations $F_\bullet K$ and $G_\bullet K_0$, and suppose that for any filtration value t we have $G_t K_0 \subseteq F_t K$. Apply the following steps:

1. Construct relative boundary matrix $\mathcal{D}$.
2. Perform a U-Match decomposition $\mathcal{T} M = \mathcal{D}\mathcal{S}$.
3. Obtain $\mathcal{A}$ and $\mathcal{B}$ by permuting columns of $\mathcal{T}$ and $\mathcal{S}$.
4. Perform a U-Match Decomposition $\mathscr{T}\mathcal{M} = (\mathcal{A}^{-1}\mathcal{B})\mathscr{S}$.

Suppose RelIm($\mathcal{D}$) has dimension i and RelKer($\mathcal{D}$) has dimension j at filtration value t.

Theorem

Let K and K_0 be simplicial complexes equipped with finite filtrations $F_\bullet K$ and $G_\bullet K_0$, and suppose that for any filtration value t we have $G_t K_0 \subseteq F_t K$. Apply the following steps:

1. Construct relative boundary matrix $\mathcal{D}$.
2. Perform a U-Match decomposition $\mathcal{T}M = \mathcal{D}\mathcal{S}$.
3. Obtain $\mathcal{A}$ and $\mathcal{B}$ by permuting columns of $\mathcal{T}$ and $\mathcal{S}$.
4. Perform a U-Match Decomposition $\mathscr{T}\mathcal{M} = (\mathcal{A}^{-1}\mathcal{B})\mathscr{S}$.

Suppose RelIm($\mathcal{D}$) has dimension i and RelKer($\mathcal{D}$) has dimension j at filtration value t. If the above steps are applied, then the set

$$COL_J(\mathcal{A}\mathscr{T}\mathcal{M}) \setminus COL_I(\mathcal{A}\mathscr{T})$$

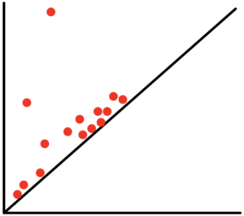
contains a basis for $H_n(F_t K, G_t K_0)$.

STABILITY

STABILITY

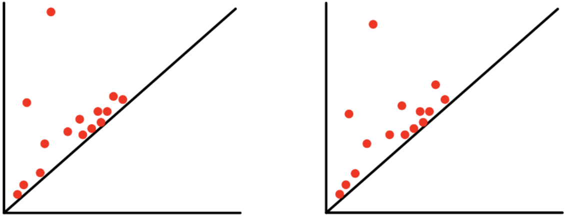
- What does it mean for a persistence algorithm to be stable?

- What does it mean for a persistence algorithm to be stable?



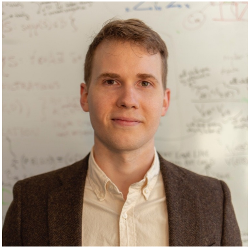
STABILITY

- What does it mean for a persistence algorithm to be stable?



- We showed that the U-Match PRH algorithm is stable using a few previously established results!

ACKNOWLEDGEMENTS

Lori Ziegelmeier	Greg Henselman-Petrusek	Xintan Xia
Macalester Col.	PNNL	Macalester Col.

THANK YOU!

ALGORITHM DETAILS

STEP 1: THE BOUNDARY MATRIX

- Assume that you start with distance matrices DK and DK_0.
- Construct a VR complex from DK to obtain $\mathcal{F}$, a filtered list of simplices corresponding to $F_\bullet K$.
- Similarly construct $\mathcal{G}$ from DK_0.
  - Use scale parameter $\varepsilon_1 \leq \cdots \leq \varepsilon_N$ for DK and scale parameter $\delta_1 \leq \cdots \leq \delta_N$ for DK_0.
  - Require that $\delta_t \leq \varepsilon_t$ for any $t \leq N$.
  - Thus, $G_t K_0 \subseteq F_t K$ for each $t \leq N$.
- Use $\mathcal{F}$ and $\mathcal{G}$ to construct relative boundary matrix $\mathcal{D}$. This is just a sorting algorithm!

STEP 3: PERMUTE COLUMNS

Do this with a sorting algorithm, using the following two algorithms as order operators.

Algorithm 1 Test Relative Cycle Birth

Require: A positive integer c which is a column index in M that corresponds to the column of $\mathcal{S}$ given by $\alpha = COL_c(\mathcal{S})$.

Ensure: Some $a \in [0, \infty)$ describing the birth of α as a relative cycle.

1: $m \leftarrow COL_c(M)$
2: $x \leftarrow b(m)$ in $G_\bullet K_0$
3: $y \leftarrow b(\alpha)$ in $F_\bullet K$
4: $a \leftarrow max(x, y)$

For step 1, note that $\mathcal{D}\alpha = \mathcal{D} \cdot COL_c(\mathcal{S}) = \mathcal{T} \cdot COL_c(M)$.

STEP 3: PERMUTE COLUMNS

Algorithm 2 Test Relative Boundary Birth

Require: A positive integer r which is a row index in M that corresponds to the column of $\mathcal{T}$ given by $\alpha = COL_r(\mathcal{T})$.

Ensure: Some $a \in [0, \infty)$ describing the birth of α as a relative boundary.

```
x ← b(α) in G•K0
m_r ← ROW_r(M)
if r ∈ r• then
    c ← index of nonzero entry in m_r
    m_c ← COL_c(M)
    y ← b(m_c) in F•K
end if
if r ∈ overline(r•) then
    y ← ∞
end if
a ← min(x, y)
```

$x \leftarrow b(\alpha)$ in $G_\bullet K_0$
$m_r \leftarrow ROW_r(M)$
if $r \in r_\bullet$ **then**
  $c \leftarrow$ index of nonzero entry in m_r
  $m_c \leftarrow COL_c(M)$
  $y \leftarrow b(m_c)$ in $F_\bullet K$
end if
if $r \in \overline{r_\bullet}$ **then**
  $y \leftarrow \infty$
end if
$a \leftarrow min(x, y)$