Feature Generation for Drug Discovery Learning

Using Persistent Homology to Create Moduli Spaces of Chemical Compounds

Anthony Bak

AYASDI

Problem Context

We want to:

- Create new drugs to solve disease

Problem Context

We want to:

- Create new drugs to solve disease
- Find new compounds to run in drug trials

Problem Context

We want to:

- Create new drugs to solve disease
- Find new compounds to run in drug trials
- Run experiments to test the inhibition properties of compounds

Problem Context

We want to:

- Create new drugs to solve disease
- Find new compounds to run in drug trials
- Run experiments to test the inhibition properties of compounds
- Find a set of compounds a small enough number to try

Problem Context

We want to:

- Create new drugs to solve disease
- Find new compounds to run in drug trials
- Run experiments to test the inhibition properties of compounds
- Find a set of compounds a small enough number to try
- Sort through all known compounds to come up with likely collection of compounds

Problem Context

We want to:

- Create new drugs to solve disease
- Find new compounds to run in drug trials
- Run experiments to test the inhibition properties of compounds
- Find a set of compounds a small enough number to try $\leftarrow$ Here is our step
- Sort through all known compounds to come up with likely collection of compounds

Problem Context

We want to:

- Create new drugs to solve disease
- Find new compounds to run in drug trials
- Run experiments to test the inhibition properties of compounds
- Find a set of compounds a small enough number to try $\leftarrow$ Here is our step
- Sort through all known compounds to come up with likely collection of compounds $\leftarrow$ Maybe with enough compute power we could do this.

Problem Context

We want to:

- Create new drugs to solve disease
- Find new compounds to run in drug trials
- Run experiments to test the inhibition properties of compounds
- Find a set of compounds a small enough number to try $\leftarrow$ Here is our step
- Sort through all known compounds to come up with likely collection of compounds $\leftarrow$ Maybe with enough compute power we could do this.

This process is called **virtual screening**

Meta Goals

- Solve the problem

Meta Goals

- Solve the problem
- Use solution to illustrate new mathematical tools. Eg. persistent homology

Meta Goals

- Solve the problem
- Use solution to illustrate new mathematical tools. Eg. persistent homology
- Tools illustrate what may be some unexpected mathematical concepts (functoriality, rings of algebraic functions etc.) being applied in a data driven (not model driven) context.

Meta Goals

- Solve the problem
- Use solution to illustrate new mathematical tools. Eg. persistent homology
- Tools illustrate what may be some unexpected mathematical concepts (functoriality, rings of algebraic functions etc.) being applied in a data driven (not model driven) context.
- Some mathematical limitations of current methods are discussed

Why do virtual screen at all?

- **High throughput screening** (HTS)
  - Physical screening of large numbers of potential drugs.
  - Very expensive

Why do virtual screen at all?

- **High throughput screening** (HTS)
  - Physical screening of large numbers of potential drugs.
  - Very expensive
- **Virtual screening**
  - Computational
  - Typically based on biochemical knowledge
  - Drastically reduces the cost of HTS
  - Typical goal for a database of millions of compounds is to select 90% of the potential inhibitors with about 10% of the total compounds.

Why do virtual screen at all?

- **High throughput screening** (HTS)
  - Physical screening of large numbers of potential drugs.
  - Very expensive
- **Virtual screening**
  - Computational
  - Typically based on biochemical knowledge
  - Drastically reduces the cost of HTS
  - Typical goal for a database of millions of compounds is to select 90% of the potential inhibitors with about 10% of the total compounds.

Many different methods:

- QSAR (quantitative structure-activity relationship)
- Pharmacophore models (points in 3D space, with radii, representing specific types of chemical interaction)
- Typically, **no insight into the space of compounds being examined**

Why do virtual screen at all?

- **High throughput screening** (HTS)
  - Physical screening of large numbers of potential drugs.
  - Very expensive
- **Virtual screening**
  - Computational
  - Typically based on biochemical knowledge
  - Drastically reduces the cost of HTS
  - Typical goal for a database of millions of compounds is to select 90% of the potential inhibitors with about 10% of the total compounds.

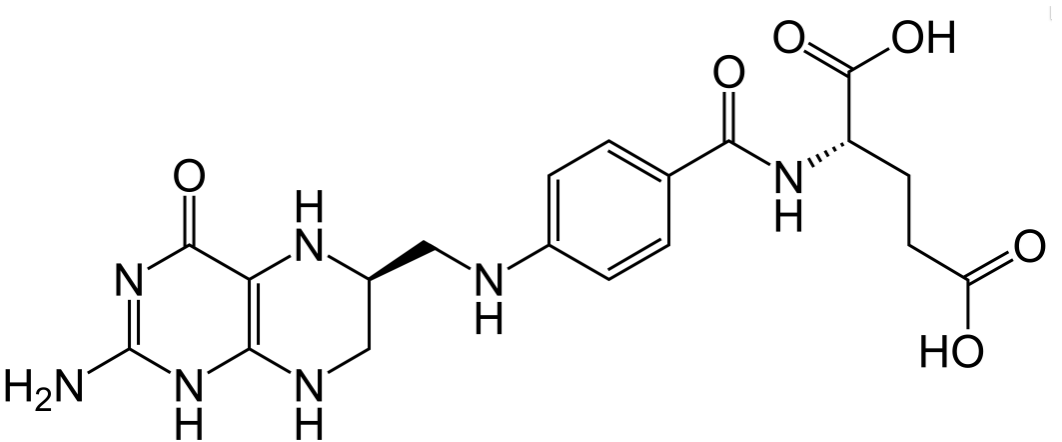
Many different methods:

- QSAR (quantitative structure-activity relationship)
- Pharmacophore models (points in 3D space, with radii, representing specific types of chemical interaction)
- Typically, **no insight into the space of compounds being examined**

Goal: To find the set of relevant bioactive compounds

Our Example: Dihydrofolate reductase (DHFR)

- Tetrahydrofolate is an important precursor in the biosynthesis of **purines**, thymidylate, and several important amino acids.
- DHFR turns dihydrofolate (DHF) into tetrahydrafolate (THF).
- Dihydrofolate is easily available. The reaction catalyzed by DHFR is the **only** source you have for THF.



Why DHFR

DHFR inhibitors are a class of drugs that stop DHFR from working. Why do we care?

- Cancer (e.g. methotrexate)
  - DNA is made from purines (**A**denine and **G**uanine) and pyrimidines (**T**hymine and **C**ytosine).
  - Stopping DHFR $\rightarrow$ no new DNA $\rightarrow$ cells cannot divide
  - Everything dies, but cancer is growing most quickly, so (hopefully) it dies first.

Why DHFR

DHFR inhibitors are a class of drugs that stop DHFR from working. Why do we care?

- Cancer (e.g. methotrexate)
  - DNA is made from purines (**A**denine and **G**uanine) and pyrimidines (**T**hymine and **C**ytosine).
  - Stopping DHFR $\rightarrow$ no new DNA $\rightarrow$ cells cannot divide
  - Everything dies, but cancer is growing most quickly, so (hopefully) it dies first.
- Bacteria (e.g. trimethoprim)
  - Bacterial DHFR has similar, but different, structure.
  - Some DHFR inhibitors only bind bacterial DHFR, not human.

Why DHFR

DHFR inhibitors are a class of drugs that stop DHFR from working. Why do we care?

- Cancer (e.g. methotrexate)
  - DNA is made from purines (**A**denine and **G**uanine) and pyrimidines (**T**hymine and **C**ytosine).
  - Stopping DHFR $\rightarrow$ no new DNA $\rightarrow$ cells cannot divide
  - Everything dies, but cancer is growing most quickly, so (hopefully) it dies first.
- Bacteria (e.g. trimethoprim)
  - Bacterial DHFR has similar, but different, structure.
  - Some DHFR inhibitors only bind bacterial DHFR, not human.
- Malaria (e.g. pyrimethamine)
  - Some DHFR inhibitors only bind malarial DHFR.

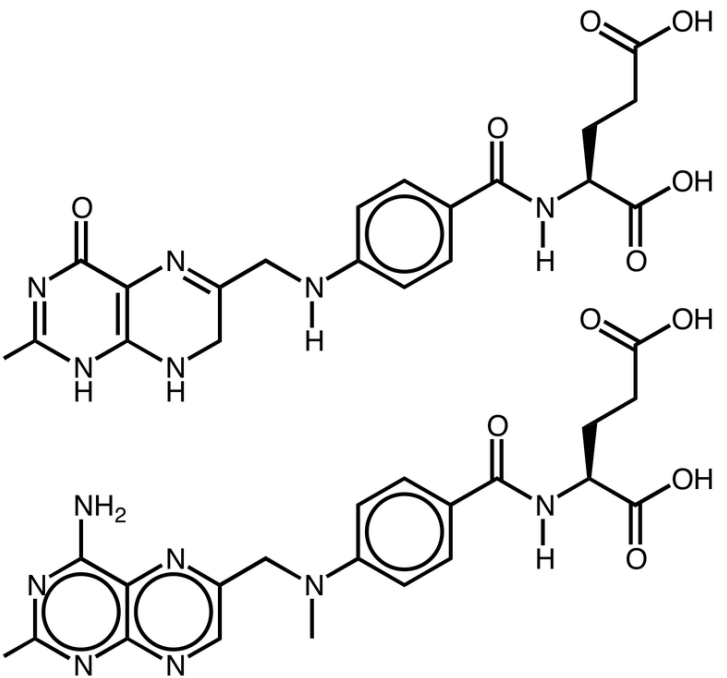
Problem Complexity

The multi-species DHFR activity makes our problem more complicated

- We need to separate out compounds not just by bioactivity but per-species bioactivity.
- You don't want a drug targeting E Coli to also function as a cancer drug that stops human cellular reproduction
- Ditto for other species pneumonia, malaria etc. so that we can have precise targeting

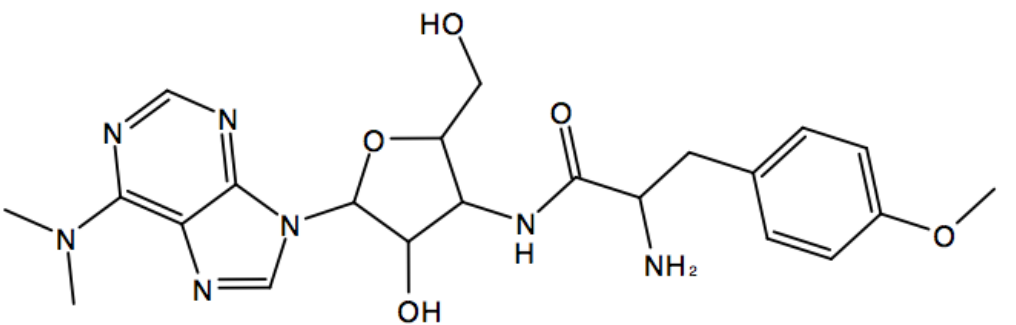
Structure-based DHFR drug design

Methotrexate, a DHFR-inhibitor, is the first historical example of successful anticancer structure-based drug design.



Structure-based DHFR drug design

For comparison, a chemically similar molecule that does *not* inhibit DHFR:



Structure-based DHFR drug design

Structure-based drug design is *hard*

- Design required significant biological and biochemical experiments and knowledge as well as years of work.

Structure-based DHFR drug design

Structure-based drug design is *hard*

- Design required significant biological and biochemical experiments and knowledge as well as years of work.
- Methotrexate, designed in the late 40's and early 50's, is still used today as an anticancer drug.

Structure-based DHFR drug design

Structure-based drug design is *hard*

- Design required significant biological and biochemical experiments and knowledge as well as years of work.
- Methotrexate, designed in the late 40's and early 50's, is still used today as an anticancer drug.
- Typical side effects: hair loss, ulcers, etc. Drugs can have bad side effects but if they're the only option...

Structure-based DHFR drug design

Structure-based drug design is *hard*

- Design required significant biological and biochemical experiments and knowledge as well as years of work.
- Methotrexate, designed in the late 40's and early 50's, is still used today as an anticancer drug.
- Typical side effects: hair loss, ulcers, etc. Drugs can have bad side effects but if they're the only option...
- Decades later, the first crystal structure of methotrexate bound to DHFR was found. It binds *upside down* in the binding pocket when compared to THF!

Structure-based DHFR drug design

Structure-based drug design is *hard*

- Design required significant biological and biochemical experiments and knowledge as well as years of work.
- Methotrexate, designed in the late 40's and early 50's, is still used today as an anticancer drug.
- Typical side effects: hair loss, ulcers, etc. Drugs can have bad side effects but if they're the only option...
- Decades later, the first crystal structure of methotrexate bound to DHFR was found. It binds *upside down* in the binding pocket when compared to THF!

Yikes!

Feature Engineering using Topology

Overview

The method:

- For each compound we calculate a set of topological invariants (barcodes)

Overview

The method:

- For each compound we calculate a set of topological invariants (barcodes)
- From a metric on barcodes make a finite metric space

Overview

The method:

- For each compound we calculate a set of topological invariants (barcodes)
- From a metric on barcodes make a finite metric space
- We hope that compounds with certain bio-chemical properties are localized in this space

Overview

The method:

- For each compound we calculate a set of topological invariants (barcodes)
- From a metric on barcodes make a finite metric space
- We hope that compounds with certain bio-chemical properties are localized in this space
- We think that with the right collection of barcodes they are.

Overview

The method:

- For each compound we calculate a set of topological invariants (barcodes)
- From a metric on barcodes make a finite metric space
- We hope that compounds with certain bio-chemical properties are localized in this space
- We think that with the right collection of barcodes they are.

Philosophy : We don't use the barcodes to study individual compounds but to say how they differ from each other. There relative differences and the global structure of the space allow us to make inferences.

Dataset

To get our test dataset we wanted to get all likely compounds:

- Based on experience we knew that each compound needed at least one aromatic group and a hydrophobic piece.
- We search a database of drug like compounds for all matching compounds
- We did a literature search for all known DHFR based drugs (across all species)
- We combined these datasets into a single dataset with 4000 compounds

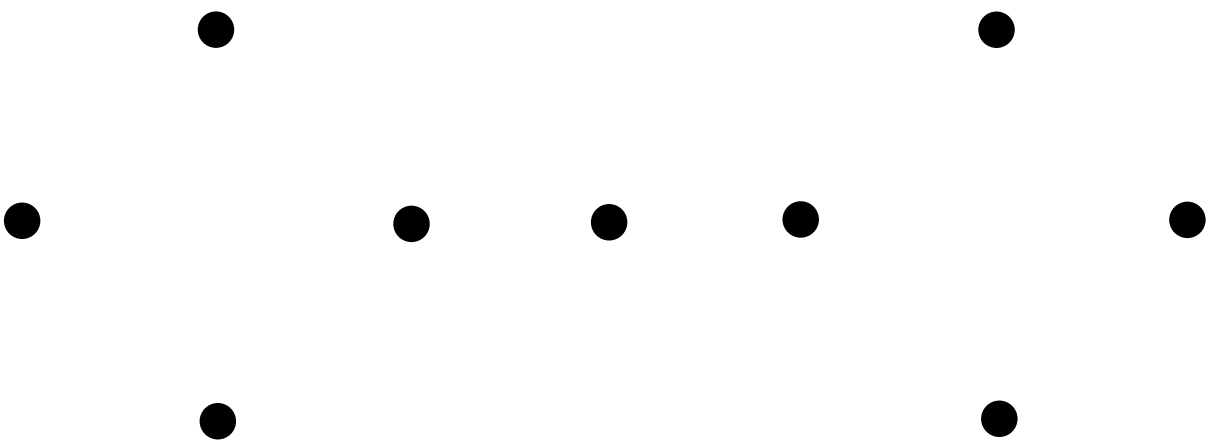
Persistent Homology: Notation and Language

Given a filtration of a space $X_0 \subseteq X_1 \subseteq X_2 ... \subseteq X$ we have maps of homology $H_i(X_l) \to H_i(X_k)$ whenever $l < k$. This situation is classified by a barcode.

- For us, filtrations will be created by sub and super level sets of functions

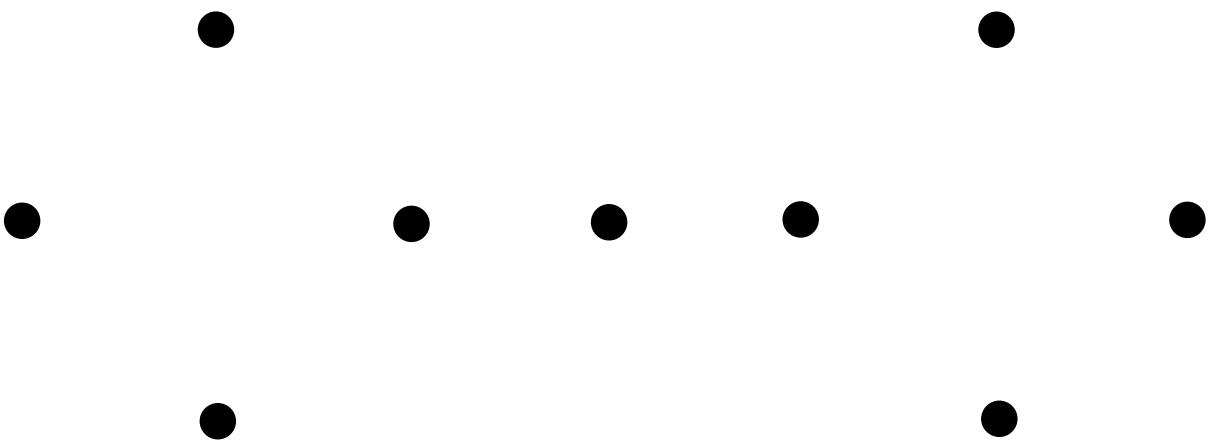
Intuition: We track when a homology class is born and when it dies according to some choice of parameter.

Persistent Homology: Rips Filtration



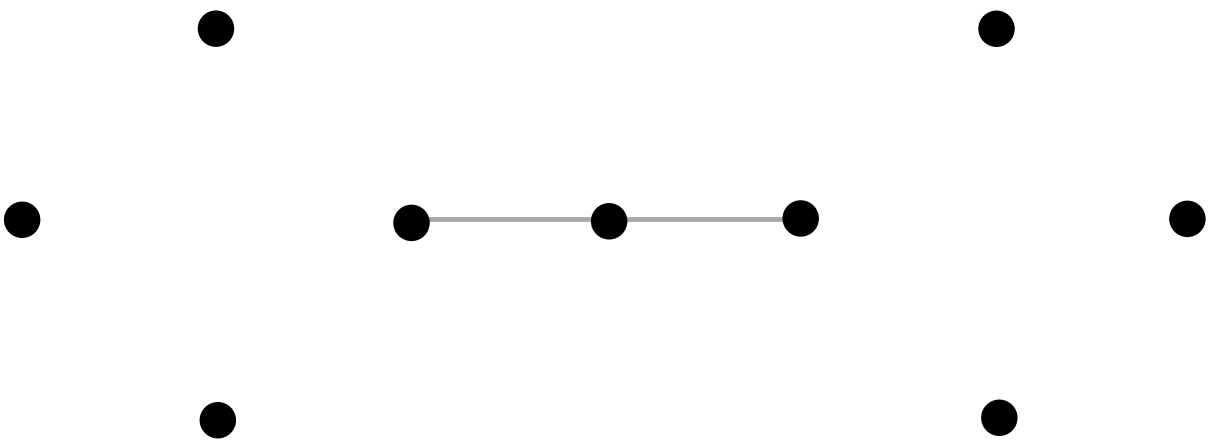
Persistent Homology: Rips Filtration

- Start with points and distances between them



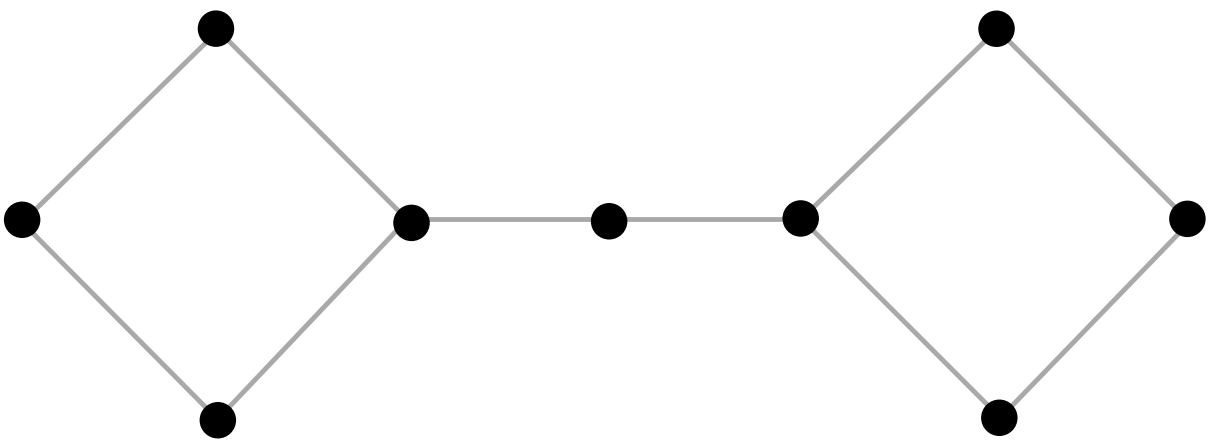
Persistent Homology: Rips Filtration

- Start with points and distances between them
- Add edges in order of increasing pairwise distances between points



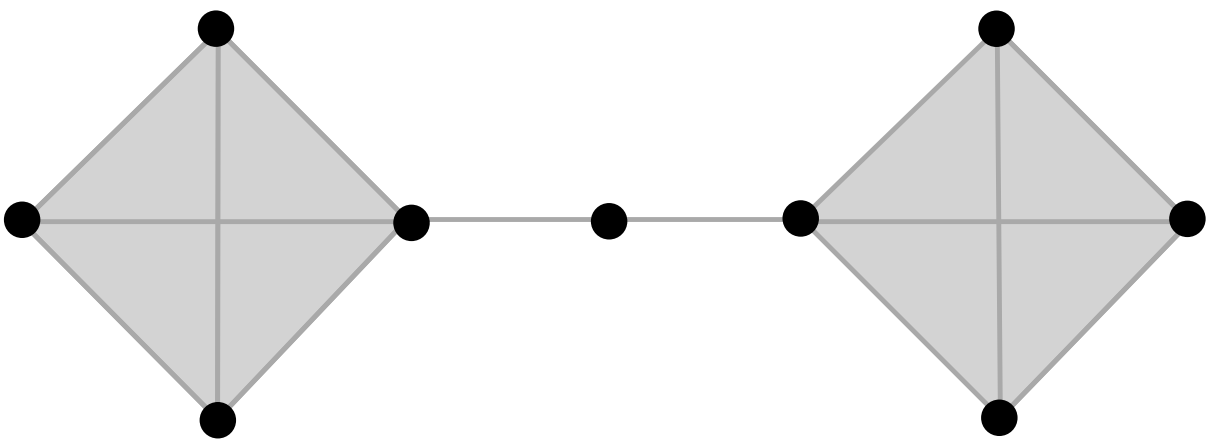
Persistent Homology: Rips Filtration

- Start with points and distances between them
- Add edges in order of increasing pairwise distances between points



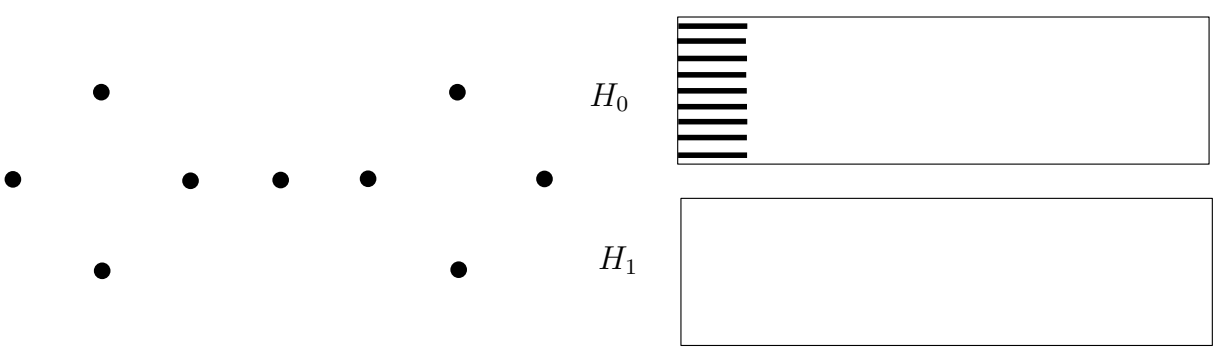
Persistent Homology: Rips Filtration

- Start with points and distances between them
- Add edges in order of increasing pairwise distances between points
- Add higher dimensional simplifies when all their faces are already included.



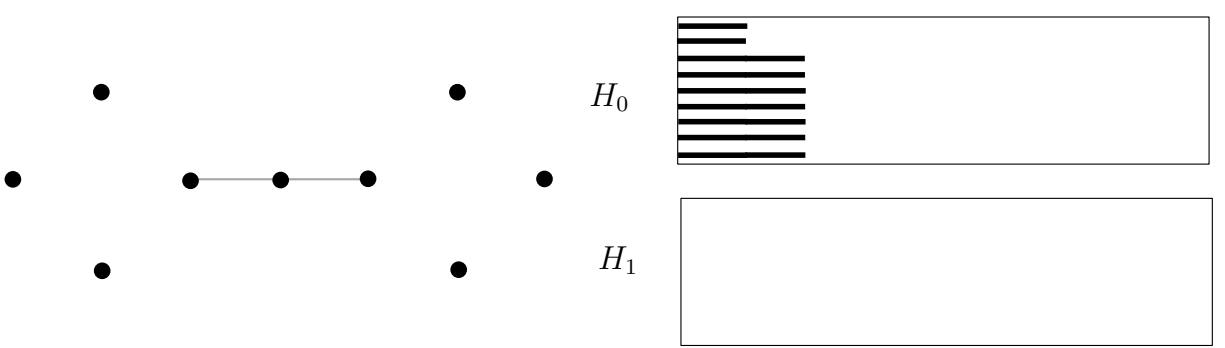
Persistent Homology: Barcodes

- Start with 9 points. Each has a barcode in dimension 0.



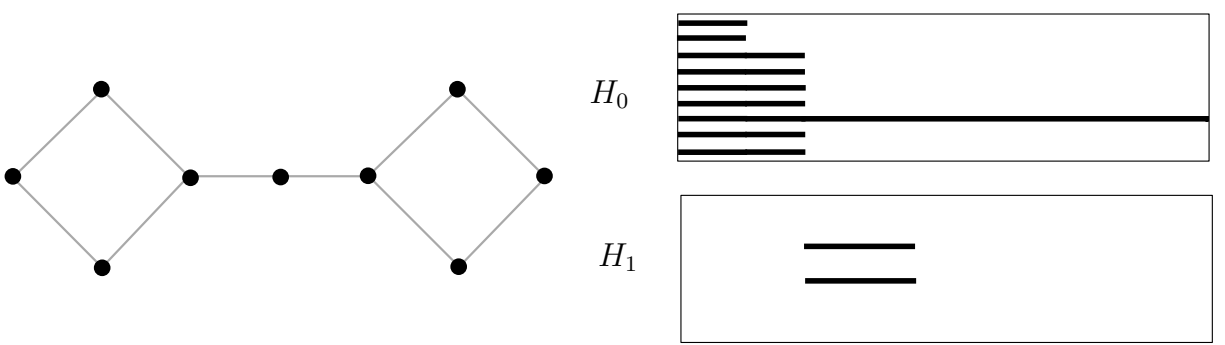
Persistent Homology: Barcodes

- Start with 9 points. Each has a barcode in dimension 0.
- Adding the shortest edges kills two components. The rest continue.



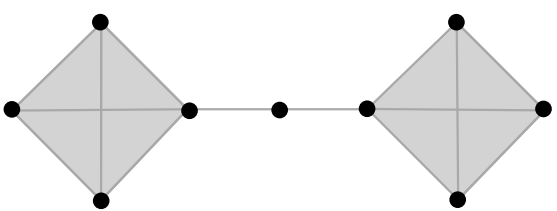
Persistent Homology: Barcodes

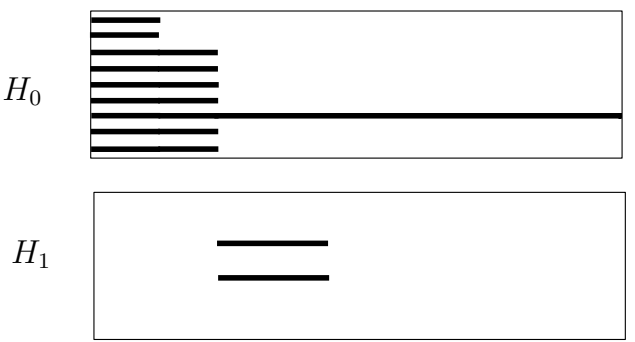
- Start with 9 points. Each has a barcode in dimension 0.
- Adding the shortest edges kills two components. The rest continue.
- We only have one component but have create two cycles in dimension 1.



Persistent Homology: Barcodes

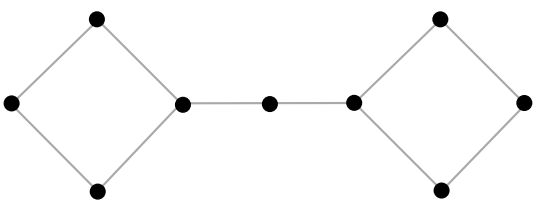
- Start with 9 points. Each has a barcode in dimension 0.
- Adding the shortest edges kills two components. The rest continue.
- We only have one component but have create two cycles in dimension 1.
- These two cycles die when the face (triangles) are filled in.





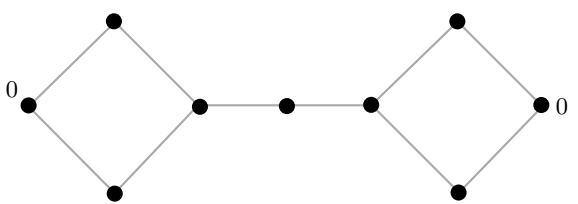
Persistent Homology: Functions

Fix an underlying complex



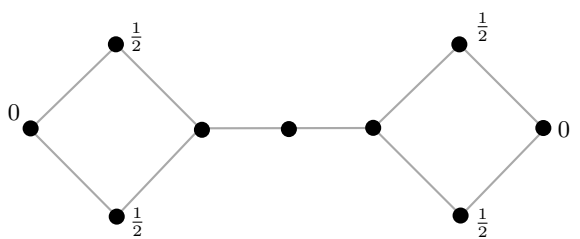
Persistent Homology: Functions

Assign each vertex a real value. This is the filtration function



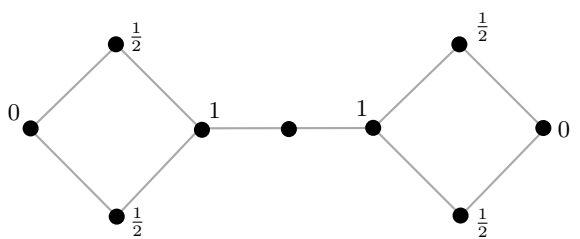
Persistent Homology: Functions

Assign each vertex a real value. This is the filtration function



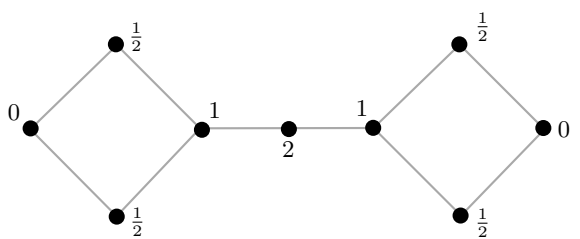
Persistent Homology: Functions

Assign each vertex a real value. This is the filtration function



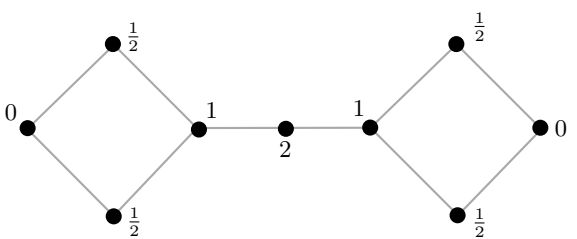
Persistent Homology: Functions

Extend the function to the higher dimensional simplicies in a linear fashion.



Persistent Homology: Functions

Track the cycles as they are created and die.

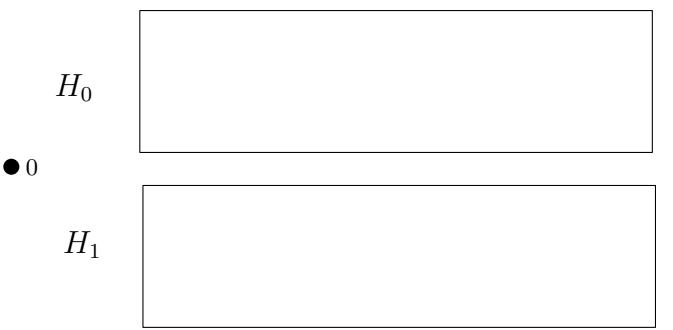


H_0

H_1

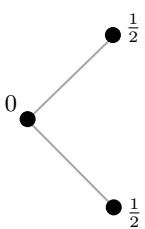
Persistent Homology: Functions

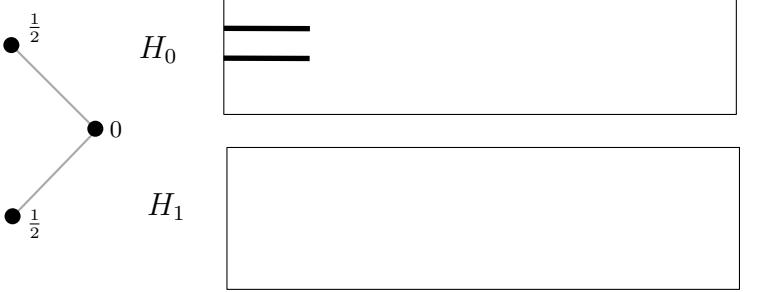
Track the cycles as they are created and die.



Persistent Homology: Functions

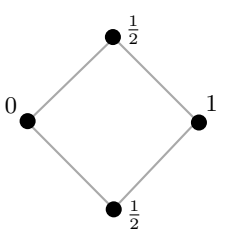
Track the cycles as they are created and die.

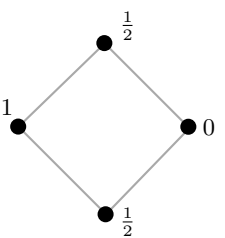


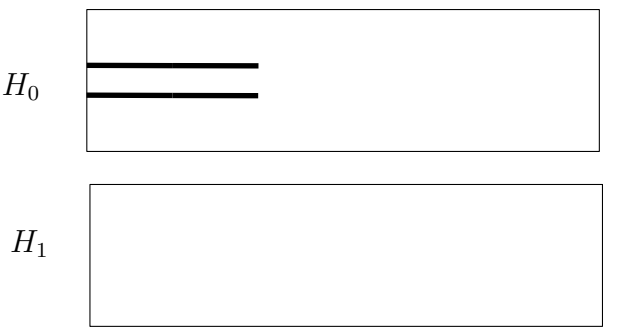


Persistent Homology: Functions

Track the cycles as they are created and die.

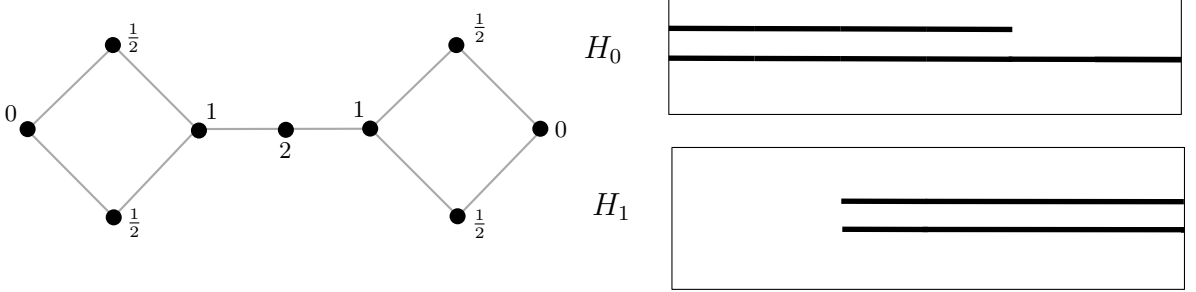






Persistent Homology: Functions

Track the cycles as they are created and die.

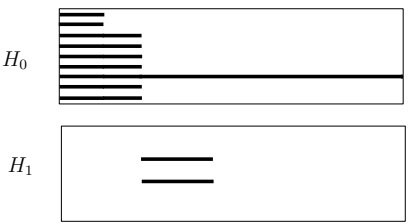


Barcode Interpretability and Engineering

- Different filtrations tell you different things about the structure.

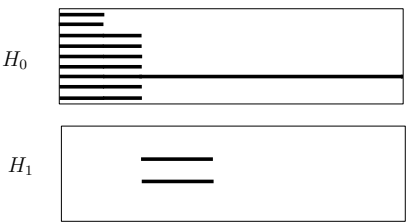
Barcode Interpretability and Engineering

- Different filtrations tell you different things about the structure.
  - For the rips filtration we learn about connected components and the size of the voids

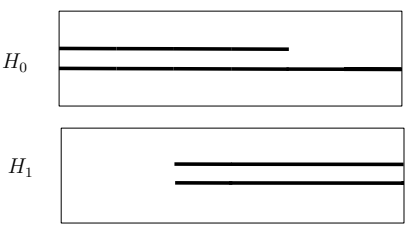


Barcode Interpretability and Engineering

- Different filtrations tell you different things about the structure.
  - For the rips filtration we learn about connected components and the size of the voids

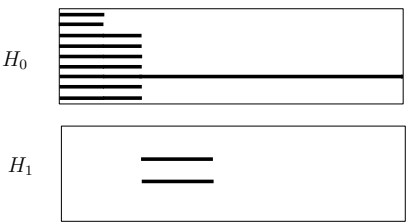


  - For the function example we learned that there are two ends and cycles away from the center

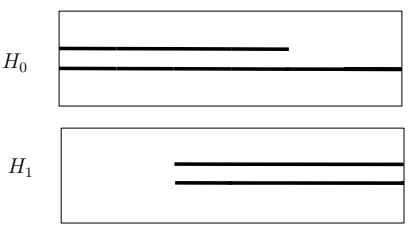


Barcode Interpretability and Engineering

- Different filtrations tell you different things about the structure.
  - For the rips filtration we learn about connected components and the size of the voids



  - For the function example we learned that there are two ends and cycles away from the center



- You can interpret the barcodes and you can engineer them to answer different questions.

Chemical Compounds as Finite Metric Spaces

A chemical compound is a finite metric space

Chemical Compounds as Finite Metric Spaces

A chemical compound is a finite metric space

- Points are 3-d atomic coordinates

Chemical Compounds as Finite Metric Spaces

A chemical compound is a finite metric space

- Points are 3-d atomic coordinates
- We considered two natural metrics
  - The embedded (euclidean) metric
  - The graph metric given by bonds and their lengths

Chemical Compounds as Finite Metric Spaces

A chemical compound is a finite metric space

- Points are 3-d atomic coordinates
- We considered two natural metrics
  - The embedded (euclidean) metric
  - The graph metric given by bonds and their lengths

A function on a compound is a real value for each atom.

- The most important property is that they have intrinsic meaning for either the chemistry or geometry.

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

- Geometric

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

- Geometric
  - α-complex filtration

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

- Geometric
  - α-complex filtration
  - Rips complex

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

- Geometric
  - α-complex filtration
  - Rips complex
  - Eccentricity

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

- Geometric
  - α-complex filtration
  - Rips complex
  - Eccentricity
- Chemical
  - Atomic mass

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

- Geometric
  - α-complex filtration
  - Rips complex
  - Eccentricity
- Chemical
  - Atomic mass
  - Partial charge

Filter functions

We want a rich set of filter functions to capture the bio-chemical properties of a compound.

- Geometric
  - α-complex filtration
  - Rips complex
  - Eccentricity
- Chemical
  - Atomic mass
  - Partial charge

Note: Even the geometric functions already capture a lot of chemical information since they are based on chemical bonds.

Some Parameter choices

For all of the filter functions except the α-complex and the rips complex a choice of scale is needed to build the underlying complex being filtered.

- We generally choose a selection of scales. Typical choices are multiples of the carbon-carbon bond length: 1,2,4,6,8.
- These are 'slices' along one direction of the multi filtration

Some Parameter choices

For all of the filter functions except the α-complex and the rips complex a choice of scale is needed to build the underlying complex being filtered.

- We generally choose a selection of scales. Typical choices are multiples of the carbon-carbon bond length: 1,2,4,6,8.
- These are 'slices' along one direction of the multi filtration

For metrics using the bond graph we need need to choose a dimension cutoff for betti numbers.

- We generally choose 2 or 3 for calculation resource reasons

Some Parameter choices

For all of the filter functions except the α-complex and the rips complex a choice of scale is needed to build the underlying complex being filtered.

- We generally choose a selection of scales. Typical choices are multiples of the carbon-carbon bond length: 1,2,4,6,8.
- These are 'slices' along one direction of the multi filtration

For metrics using the bond graph we need need to choose a dimension cutoff for betti numbers.

- We generally choose 2 or 3 for calculation resource reasons

Some Parameter choices

For all of the filter functions except the α-complex and the rips complex a choice of scale is needed to build the underlying complex being filtered.

- We generally choose a selection of scales. Typical choices are multiples of the carbon-carbon bond length: 1,2,4,6,8.
- These are 'slices' along one direction of the multi filtration

For metrics using the bond graph we need need to choose a dimension cutoff for betti numbers.

- We generally choose 2 or 3 for calculation resource reasons

For the rips filtration we need to choose a maximum distance parameter.

- We generally choose 6 or 8 times the carbon-carbon bond length for calculation resource reasons

Barcode Zoo

There's a combinatorial explosion of parameters and resulting barcodes

- We end up with hundreds of barcodes for each compound.

For linguistic reasons to match the language from computational chemistry we might call them **topological fingerprints**.

Compounds to Metric Spaces

For each barcode we use the either the bottleneck or Wasserstein distance to form a metric space of compounds.

Bottleneck:

$$B(B_1, B_2) = \inf_{m: B_1 \to B_2} (\sup_{b \in B_1} ||b - m(b)||_\infty)$$

Wasserstein:

$$W(B_1, B_2) = \inf_{m: B_1 \to B_2} (\sum_{b \in B_1} ||b - m(b)||_\infty^q)^{\frac{1}{q}}$$

where m is a matching between the diagrams.

Compounds to Metric Spaces

For each barcode we use the either the bottleneck or Wasserstein distance to form a metric space of compounds.

Bottleneck:

$$B(B_1, B_2) = \inf_{m: B_1 \to B_2} (\sup_{b \in B_1} ||b - m(b)||_\infty)$$

Wasserstein:

$$W(B_1, B_2) = \inf_{m: B_1 \to B_2} (\sum_{b \in B_1} ||b - m(b)||_\infty^q)^{\frac{1}{q}}$$

where m is a matching between the diagrams.

- Both are kinds of "edit" distances. For bottleneck we take the largest edit in the best matching while for Wasserstein we sum the edits for each bar.

Compounds to Metric Spaces

For each barcode we use the either the bottleneck or Wasserstein distance to form a metric space of compounds.

Bottleneck:

$$B(B_1, B_2) = \inf_{m: B_1 \to B_2} (\sup_{b \in B_1} ||b - m(b)||_\infty)$$

Wasserstein:

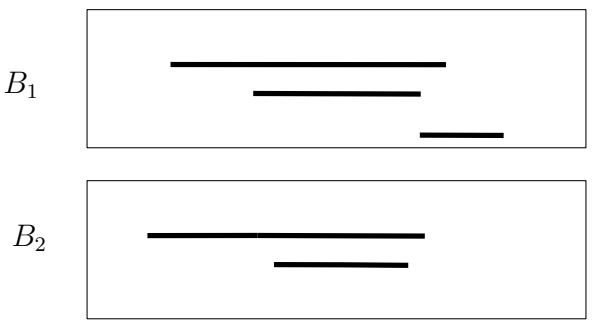
$$W(B_1, B_2) = \inf_{m: B_1 \to B_2} (\sum_{b \in B_1} ||b - m(b)||_\infty^q)^{\frac{1}{q}}$$

where m is a matching between the diagrams.

- Both are kinds of "edit" distances. For bottleneck we take the largest edit in the best matching while for Wasserstein we sum the edits for each bar.
- We allow bars to be matched to 'zero' as well to make this work

Matching Distances

The distances are an intuitive way to understand how two barcodes differ.

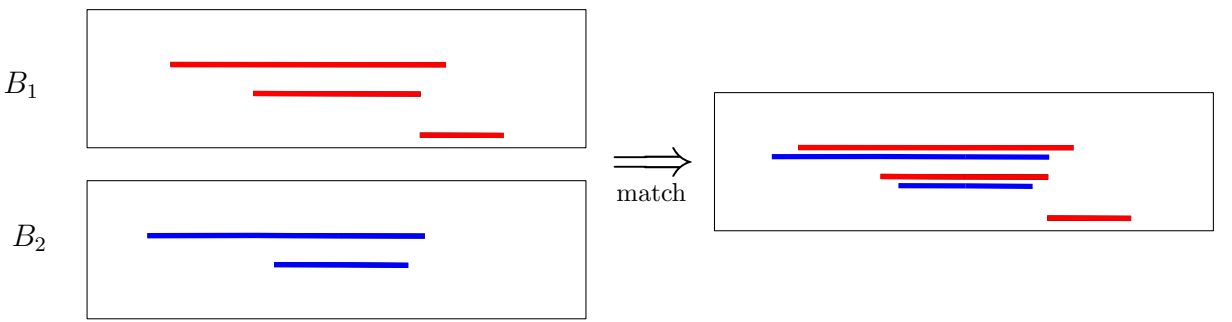


Matching Distances

The distances are an intuitive way to understand how two barcodes differ.

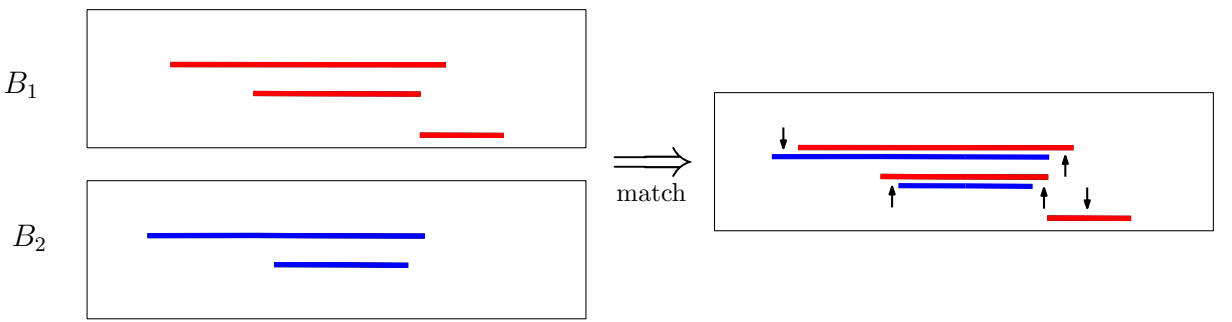
Matching Distances

The distances are an intuitive way to understand how two barcodes differ.



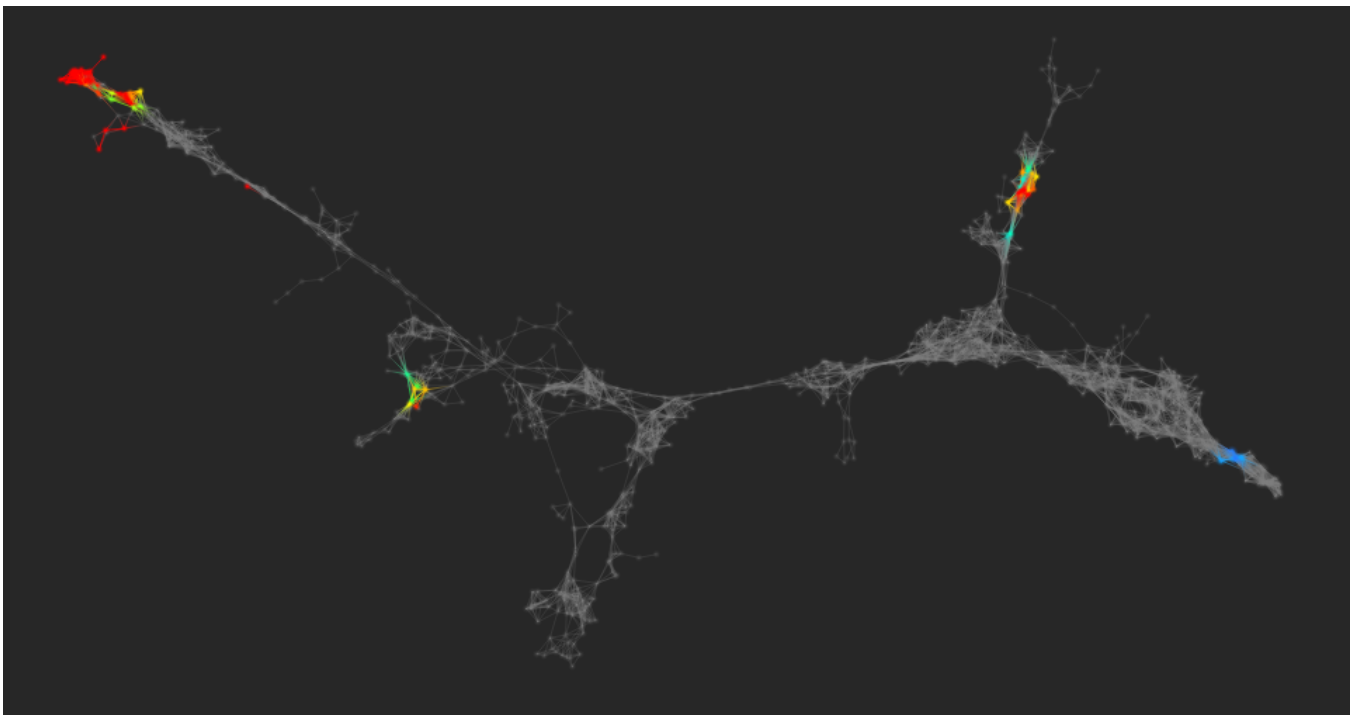
Matching Distances

The distances are an intuitive way to understand how two barcodes differ.



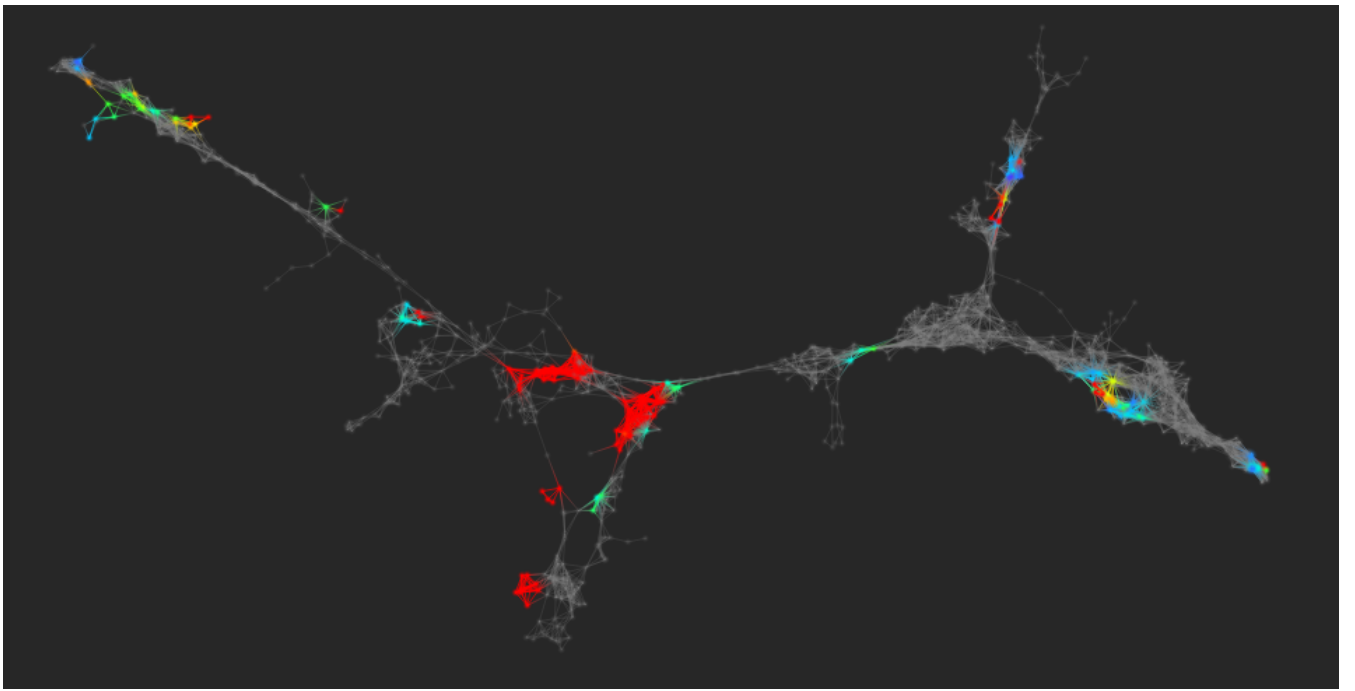
Visualization and Discovery: Ayasdi Mapper

Known Human Inhibitors



Visualization and Discovery: Ayasdi Mapper

Known E. Coli Inhibitors



Machine Learning: Functions and SVM

The space of barcodes forms an algebraic variety

Machine Learning: Functions and SVM

The space of barcodes forms an algebraic variety

Machine Learning: Functions and SVM

The space of barcodes forms an algebraic variety

- We know the ring of functions. Writing writing (x_i, y_i) for a (birth,death) point in a barcode some examples are:

$$\sum(y_i - x_i)$$
$$\sum(y_i - x_i)^2 \quad \sum(y_i - x_i)(x_i + y_i)$$

Note that $\sum x_i + y_i$ is not in the ring since want functions to be zero on diagrams with only length zero bars.

Machine Learning: Functions and SVM

The space of barcodes forms an algebraic variety

- We know the ring of functions. Writing writing (x_i, y_i) for a (birth,death) point in a barcode some examples are:

$$\sum(y_i - x_i)$$
$$\sum(y_i - x_i)^2 \quad \sum(y_i - x_i)(x_i + y_i)$$

  Note that $\sum x_i + y_i$ is not in the ring since want functions to be zero on diagrams with only length zero bars.

- We can use this to embed our compound space into euclidean space and then have access to many standard machine learning algorithms. Eg. SVM classifications.

Machine Learning: Functions and SVM

The space of barcodes forms an algebraic variety

- We know the ring of functions. Writing writing (x_i, y_i) for a (birth,death) point in a barcode some examples are:

$$\sum(y_i - x_i)$$
$$\sum(y_i - x_i)^2 \quad \sum(y_i - x_i)(x_i + y_i)$$

  Note that $\sum x_i + y_i$ is not in the ring since want functions to be zero on diagrams with only length zero bars.
- We can use this to embed our compound space into euclidean space and then have access to many standard machine learning algorithms. Eg. SVM classifications.
- We do this using all polynomials up to a fixed degree.

Machine Learning: Functions and SVM

The space of barcodes forms an algebraic variety

- We know the ring of functions. Writing writing (x_i, y_i) for a (birth,death) point in a barcode some examples are:

$$\sum(y_i - x_i)$$
$$\sum(y_i - x_i)^2 \quad \sum(y_i - x_i)(x_i + y_i)$$

  Note that $\sum x_i + y_i$ is not in the ring since want functions to be zero on diagrams with only length zero bars.
- We can use this to embed our compound space into euclidean space and then have access to many standard machine learning algorithms. Eg. SVM classifications.
- We do this using all polynomials up to a fixed degree.
- Now use standard support vector machine.

SVM for Classification

SVM Confusion Matrix for E Coli, Human, C Albicans and P Carinii DHFR inhibitors:

$$\begin{bmatrix} 101 & 2 & 0 & 3 \\ 0 & 71 & 0 & 0 \\ 3 & 0 & 256 & 17 \\ 1 & 0 & 25 & 299 \end{bmatrix}$$

$\Longrightarrow$ This result is comparable to state of the art computational chemistry fingerprint and simulation based methods.

Summary

Overview

- From a set of chemical compounds calculate a rich set of barcodes
- Use a barcode metric to form the compounds into a metric space
- Understand the structure of the resulting space, for eg. via known drugs.

Summary

Overview

- From a set of chemical compounds calculate a rich set of barcodes
- Use a barcode metric to form the compounds into a metric space
- Understand the structure of the resulting space, for eg. via known drugs.

Computational topology

- Achieves state of the art accuracy for classification
- **Provides a global view of a space inaccessible previously**

Improvements

Math:

- Multidimensional Persistence: Ideally we would do all filters simultaneously.
  - Fewer parameters to choose arbitrarily.
  - Understand how the different filtrations interact.
- Optimization of barcode combinations: What do we do with the barcode zoo?

Improvements

Math:

- Multidimensional Persistence: Ideally we would do all filters simultaneously.
  - Fewer parameters to choose arbitrarily.
  - Understand how the different filtrations interact.
- Optimization of barcode combinations: What do we do with the barcode zoo?

Computer Science:

- Faster more memory efficient persistence homology calculations

Improvements

Math:

- Multidimensional Persistence: Ideally we would do all filters simultaneously.
  - Fewer parameters to choose arbitrarily.
  - Understand how the different filtrations interact.
- Optimization of barcode combinations: What do we do with the barcode zoo?

Computer Science:

- Faster more memory efficient persistence homology calculations

Chemisty:

- More domain specific filters. Eg. Color filtrations.
- Weighted versions of filters we have

Acknowledgements

- Joint work with Michael G. Lerner, Earlham College department of Physics and Astonomy.
- Sponsoring Institutions: American Institute of Mathematics, Stanford University, Ayasdi, National Institutes of Health, Earlham College.
- Calculation of Persistent Homology done with Dionysus http://www.mrzv.org/software/dionysus/. Thanks Dmitriy Morozov!