

Invariants and Metrics
for
Multiparameter Persistent Homology

by

Ashleigh Linnea Thomas

Department of Mathematics
Duke University

Date: _____

Approved:

Ezra Miller, Supervisor

Paul Bendich

John Harer

Sayan Mukherjee

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
Duke University

2019

ABSTRACT

Invariants and Metrics for Multiparameter Persistent Homology

by

Ashleigh Linnea Thomas

Department of Mathematics
Duke University

Date: _____

Approved:

Ezra Miller, Supervisor

Paul Bendich

John Harer

Sayan Mukherjee

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
Duke University

2019

Copyright © 2019 by Ashleigh Linnea Thomas

All rights reserved

Abstract

This dissertation is about building fundamental techniques for comparing data via a geometric and topological data analysis method called multiparameter persistent homology. The techniques used are largely algebraic. A new summary statistic, called the multirank function, is introduced as a measure of persistence output that detects relationships between important features of the data being analyzed. Also introduced is a technique for modifying existing metrics on the space of persistence outputs. Existing metrics can return infinite distances, which do not give as much information as a finite distance; the proposed modification gives fewer such situations. The final chapter of this dissertation details work in a long-term biology research project. Persistence is used to study the relationship between continuous morphological variation and rates of topologically abnormal morphologies in populations of fruit flies. Some preliminary computations showing proof of concept are included. Future plans involve using theoretical contributions from this dissertation for final analysis of the fly data.

The distance modification is joint work with Ezra Miller and the biology application is joint with Surabhi Beriwal, Ezra Miller, and biologists at the Houle Lab at Florida State University.

To Paul and Bryn.

Contents

Abstract	iv
List of Figures	x
List of Abbreviations	xii
Acknowledgements	xiii
1 Introduction	1
2 Background: Persistence	4
2.1 Intuition and applications	4
2.2 Limitations	5
2.3 Definition of persistence	6
2.3.1 Poset modules	7
2.3.2 Persistence	10
2.3.3 Persistence translation: diagrams/barcodes $\leftrightarrow$ modules	12
2.4 Tameness and poset module encodings	14
3 Invariants and summary statistics	18
3.1 Known invariants	18
3.2 The multirank invariant	22
3.2.1 Definition	23
3.2.2 Properties	26
3.2.3 Intuition	30
3.3 Geometric invariants from multirank	33
3.3.1 Births and deaths	34
3.3.2 Merge and branch numbers	37
3.4 Computation	41
3.4.1 m -rank: a pruned multirank	41

3.4.2	Example computations	44
4	Metrics	45
4.1	Known metrics	46
4.1.1	Interleaving distance	47
4.1.2	Matching distance	50
4.1.3	Existing L_p distances	53
4.2	New L_p metrics	55
4.2.1	Qualitative features of L_p distances	55
4.2.2	L_p m -rank metric	57
4.3	Primary decomposition distances	58
4.3.1	Primary decomposition	59
4.3.2	Extracting components of a module	63
4.3.3	Metrics and norms	66
4.3.4	Primary-component-wise norms and distances	68
4.3.5	The graph and rank modules	71
5	Application: topological evolution in fly wing venation	74
5.1	Project goals	76
5.1.1	Short-term goals	77
5.1.2	Long-term goals	79
5.2	Biological context	79
5.2.1	Existing techniques for quantifying shape	79
5.2.2	Biological hypotheses	81
5.3	Mathematical setting	82
5.3.1	The two-parameter filtration	82
5.3.2	Why this filtration?	85
5.3.3	Mathematical conveniences for computation	86

5.4	Metrics	87
5.4.1	Stability	88
5.4.2	Metrics on fly wings	90
5.4.3	Metrics on persistence modules	90
5.5	Software	91
5.5.1	Biologist-developed preprocessing	92
5.5.2	Piecewise-linear fly wing approximations with error bounds . .	93
5.5.3	Filtration visualization with connected-component homology estimation	95
5.6	Conclusion and preliminary results	96
6	Conclusion	103
	References	110
	Biography	111

List of Figures

1.1	Persistent homology pipeline.	2
2.1	The shape of data looks different at different scales.	4
2.2	Covering relations, indegree, and outdegree.	8
2.3	Diagrammatic representations of poset modules.	9
2.4	A morphism between poset modules.	10
2.5	Comparing persistence diagrams, barcodes, and modules.	13
2.6	Encoding a module over a finite poset.	15
2.7	An encoding.	17
3.1	Rank does not detect births and deaths.	21
3.2	$\mathbb{Z}^n$ -module presentations: Rank does not detect births and deaths.	22
3.3	Pictorial description and example of multirank.	25
3.4	Restricting multirank to sets $S \preceq T$ loses information.	27
3.5	The multirank invariant is not complete on the space of tame poset modules.	29
3.6	Computing multiranks of a free rank 1 module.	32
3.7	“Missing” morphisms that prevent multirank from matching intuition.	33
3.8	Matlis duality relates the top (births) and socle (deaths) of a $\mathbb{Z}^n$ -module.	36
3.9	Merge numbers measure the interactivity of features.	38
4.1	Diagrammatic definition of ϵ -interleaving.	48
4.2	Bottleneck vs. L_p rank distances on single-persistence barcodes.	57
4.3	Ideals and their corresponding faces.	60
4.4	A module and a minimal primary decomposition.	62
4.5	A module with its localizations and global supports.	64
4.6	Computing reduced local support along a face of a module.	66
4.7	Examples of $\mathbb{R}_{\geq 0}^2$ modules and their 1-lengths.	69

5.1	Examples of topological variation in fly wings.	75
5.2	Continuous variation in fly wings: locations of <i>landmarks</i>	75
5.3	Two-parameter fly wing filtration.	84
5.4	A toy model fly wing with its filtration and a finite encoding.	84
5.5	How the fly wing filtrations measures geometric information.	86
5.6	Effects of stratification on stability of persistence.	89
5.7	A fly wing image beside approximations of the wing veins.	92
5.8	Fly wing image preprocessing.	93
5.9	Splines fitted to the veins of a wing are segmented into quadratic curves.	94
5.10	Fly wing splines, their linear approximations, and landmarks.	95
5.11	Approximate Hilbert functions of persistence fly wing modules.	98
5.12	Absolute differences between simulated wing Hilbert functions.	99
5.13	Pairwise L_1 -Hilbert distances (definition 4.3.26) between simulated wings.	99
5.14	Hilbert function differences: wing with extra cross-vein vs. others.	101
5.15	Hilbert function differences: continuously deformed wing vs. others.	101
5.16	Fly wing filtration: high s and $r < s$ give single-persistence on the vertices.	102

List of Abbreviations

(r, s)	parameters for fly wing filtration and persistence module
$ \cdot $	absolute value
coker	cokernel
$\cong$	isomorphic
Hilb	Hilbert function
ker	kernel
$\mathbb{k}$	the coefficient field for homology computations
$\mathbb{N}$	natural numbers (including 0)
$\ \cdot\ $	norm, length
$\prec$	strictly precedes
$\preceq$	precedes or is equal to
$\mathbb{R}$	real numbers
σ	a face of a polyhedral partially ordered group
soc	vector space generated by the cogenerators (deaths) of a module
$\succ$	strictly succeeds
$\succeq$	succeeds or is equal to
τ	a face of a polyhedral partially ordered group
$\overline{\tau}$	a face that is the complement of τ in a polyhedral partially ordered group
top	vector space corresponding to the generators (births) of a module
$\mathbb{Z}$	integers
P	a poset; usually also a parameter space for persistent homology
Q	sometimes a poset or a polyhedral partially ordered group; usually also a parameter space for persistent homology

Q_+	the positive cone of a (polyhedral) partially ordered group Q
multipersistence	multiparameter persistent homology
PCA	principal component analysis
persistence	persistent homology
poset	partially ordered set
single-persistence	single-parameter persistent homology

Acknowledgements

To my partner Paul: thank you for being with me through the journey.

Thank you to my advisor Ezra and my committee Sayan, Paul, and John.

Thank you to Surabhi for being a wonderful collaborator. Thank you to David Houle and current and previous members of the Houle Lab at Florida State University for providing data and collaborating with us.

Thank you to the National Science Foundation for funding many of my years in graduate school.

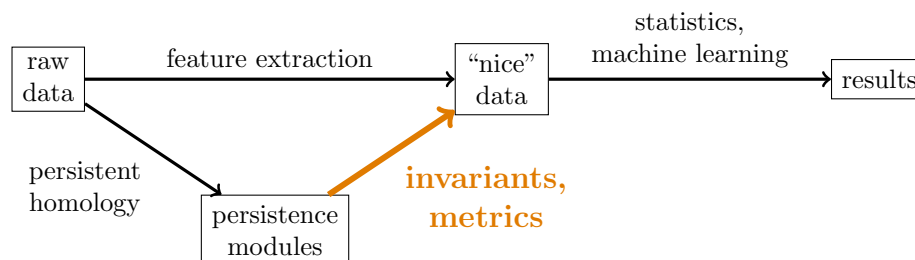
Chapter 1

Introduction

Feature extraction is a name for methods that aim to pull important, useful information from data. More specifically, feature extractors attempt to

- 1) discard extraneous information,
- 2) record useful information in an efficient way that is easy to work with statistically, and
- 3) present information in a way that is interpretable for applications.

An example of a feature extractor is *persistent homology*, a geometric and topological data analysis technique that studies the shape of data. Persistent homology extracts information about shape at different scales and assembles that information into an object that shows how shape-related features change as the scale changes. This output has already “forgotten” most of the information not related to shape and can be presented in a way that is interpretable to non-mathematicians, but it is not easy to work with statistically. This dissertation is about how to modify persistence output so that it can be fed into statistical analysis and machine learning algorithms. The tools for these modifications are invariants and metrics.



There are two distinct ways to apply persistence to a problem: An entire data set can be summarized by one persistence module, or each data object can be summarized by its own persistence module. The former tactic is useful when there are many measurements of the same object and the goal is to get information about that object’s shape, like when measuring many points inside a single 3-D object to construct a digital representation of it. The latter tactic is useful when there are many objects being compared, like in a biological experiment with many specimens. This dissertation focuses on the second setting and therefore has a focus on quantitatively comparing persistence outputs.

Persistence output is in the form of a *persistence module* (definition 2.3.14). Persistence modules do not fit well into a statistical framework on their own, but there are properties of these modules that can be measured, compared, and fed into statistics algorithms. These properties, which are called *invariants* or *summary statistics*, are also frequently used to create metrics on the space of persistence modules. It is these invariants and pairwise distances that are the statistically amenable outputs of persistence.

Organization

Chapter 2 introduces persistent homology and includes applications and limitations of persistence. The definitions presented may be unfamiliar to some researchers, scientists, and engineers who work with persistence since some use different objects to talk about results of persistent homology algorithms. In acknowledgment of these different interpretations, the persistence definitions are accompanied by an explicit translation between a common presentation of persistence called *persistence diagrams* and the module-theoretic point of view presented by the author. This translation is in subsection 2.3.3.

Following the background are chapters 3 and 4, which describe existing invariants and metrics for persistence as well as some novel ones. These chapters contain the main theoretical contributions of this dissertation. The *primary decomposition metric modification* in section 4.3 is joint work with Ezra Miller.

The final (non-concluding) chapter – chapter 5 – describes work from an ongoing collaboration with Surabhi Beriwal, Ezra Miller, and biologist David Houle. The group is analyzing data of fly wings using persistence: each wing is to be summarized by a persistence module to fuse measurements of physical characteristics resulting from both continuous and discrete evolutionary changes into one measurement. Results from chapters 3 and 4 are used to find preliminary results in section 5.6 and are part of the plans for final analysis as described in subsection 5.4.3. This application of persistence is used as a running example throughout the text.

Chapter 2

Background: Persistence

Persistent homology is a geometric data analysis tool that assigns quantitative measurements (*persistence*) to qualitative, topological features of data. Essentially, it is a way to measure some aspects of the shape of data.

2.1 Intuition and applications

Some problems with analyzing the shape of data are:

- 1) Data is noisy and to analyze shape one needs to determine what constitutes noise. One could choose a threshold for noise, but sometimes analysis can change drastically with different choices of threshold. For example, in figure 2.1 what looks like a noisy point at one scale ends up looking like a circle at another scale.

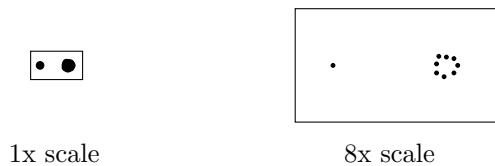


Figure 2.1. The shape of data looks different at different scales.

- 2) To compare the shapes of two objects, many methods require that the objects first be aligned with one another, which brings in theoretical and computational issues about what constitutes an optimal or sufficient alignment.

- 3) Some algorithms that compare object shapes assume that each data point fits a certain prototype, and in particular has the same topology as and is a continuous deformation of a standard example. This is the problem with many techniques for studying wing venation in fruit flies; for more details see subsection 5.2.1.

Persistent homology is a 1) multiscale, 2) coordinate-independent method to 3) quantify the shape of data with no major assumptions on input data.

Persistent homology detects qualitative features of data like the presence of connected components (e.g. the clusters of data points in [LK+12]), “holes” (e.g., the tunnels, voids, or obstacles in [BGK15]), or repetition (e.g. the patterns or regular structures in [LB+17]). To each feature, persistent homology attaches a measure of importance, robustness, or likelihood called the *persistence* that is related to the range of scales at which that feature is present.

2.2 Limitations

Persistent homology is limited because it

- 1) can identify important geometric features but does not isolate their locations,
- 2) is unstable to outliers in some of its most common application settings, and
- 3) is computationally intensive.

Limitation 1 is being remedied by algorithms that find shortest or otherwise “nicest” representatives for homology classes [Oba18].

Limitation 2 is a problem from *single-parameter* persistence [BGH11]. In that setting

there is an adaptation of persistence called *distance to a distribution* [CCM11] that takes density of data into account to make it robust to some number of outliers, but it also adds another parameter into the system. In [CI+08] Carlsson, Ishkhanov, et al. suggest that the outlier stability problem could be fixed using *multiparameter* persistence. Multiparameter persistence, or *multipersistence*, is the major topic of study for this dissertation.

Limitation 3 is the target of ongoing research and has been improved by software such as RIVET [KLW14; LW15], Ripser [Bau15], and Eirene [Hen16; HG16]. Finding new invariants and metrics that can be computed without computing the whole persistence module output is another way to reduce computational cost; this is possible future work related to the material in chapter 3.

2.3 Definition of persistence

The output or result of running a persistent homology algorithm is formulated in terms of modules. The modules in question are called *poset modules* (definition 2.3.5) and are described in the first subsection. The second subsection describes how the persistent homology algorithm works, i.e. how to transform input data into persistence modules, and sets some notation. The last subsection relates other well-known representations of persistence – bar codes and persistence diagrams – to this module-centric point of view.

2.3.1 Poset modules

Definition 2.3.1. A *poset* is a set P with a binary relation $\preceq$, the *partial order*, that is reflexive, antisymmetric, and transitive. Any $\mathbf{p} \in P$ may be referred to as a *degree* or a *point* (we reserve the term “element” for a different object).

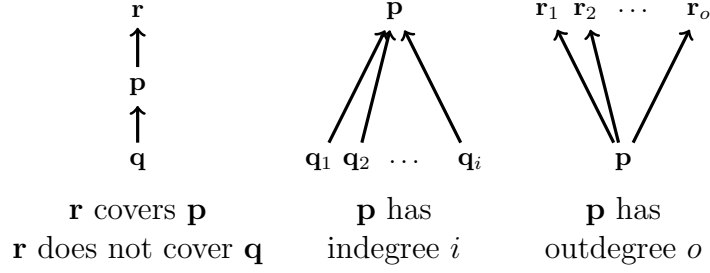
Notation. We will also use the notation $\succeq$, $\prec$, and $\succ$ to describe the partial order of a poset. $\prec$ and $\succ$ are both strict.

Motivation 2.3.2 (Relevance of posets to persistence). Posets arise in persistent homology as *parameter spaces* (see first paragraph of subsection 2.3.2). The input to persistence is a filtration of a topological space (definition 2.3.13) and that filtration is indexed by a poset: every filtered piece is associated to a point in the poset and every inclusion of filtered pieces is associated to an arrow or relation. The same poset ends up being the organizing structure for the resulting persistence module.

Notation. Points in a poset are written in bold. The two posets that are most frequently used as parameter spaces for persistence are $\mathbb{Z}^n$ and $\mathbb{R}^n$. Points in those posets are vectors or tuples and are sometimes denoted in bold; the choice to bold all points in posets is made to hint at this example.

Definition 2.3.3. Fix a locally finite poset P with points $\mathbf{p}, \mathbf{q}, \mathbf{r} \in P$.

- $\mathbf{p}$ *covers* $\mathbf{q}$ if there is no $\mathbf{r}$ such that $\mathbf{q} \prec \mathbf{r} \prec \mathbf{p}$.
- The *indegree* of $\mathbf{p}$ is the number of points that $\mathbf{p}$ covers.
- The *outdegree* of $\mathbf{p}$ is the number of points that cover $\mathbf{p}$.



Indegree and outdegree are not standard terms in this context.

For a finite poset P the transitive closure of the covering relation gives the partial order. We frequently represent posets diagrammatically by their covering relation instead of their partial order because the former leaves all identity and composition relations implicit. Such a diagram is called a Hasse diagram.

Definition 2.3.4. A *Hasse diagram* of a finite poset P is a visual representation of the covering relation of P . Each point $\mathbf{p} \in P$ corresponds to a vertex in the visualization and each cover $\mathbf{p} \preceq \mathbf{q}$ corresponds to an arrow from the vertex of $\mathbf{p}$ to the vertex of $\mathbf{q}$. The vertices are usually arranged so all the arrows point “up.” Many finite posets in this manuscript are represented using Hasse diagrams.

Definition 2.3.5. Suppose P is a poset with partial order $\preceq$. A P -module M is a collection of vector spaces and commuting linear maps indexed by P . That is,

- $M = \bigoplus_{\mathbf{p} \in P} M_{\mathbf{p}}$, where each $M_{\mathbf{p}}$ is a $\mathbb{k}$ -vector space;
- for every $\mathbf{p} \preceq \mathbf{q}$ in P , $M_{\mathbf{p}} \rightarrow M_{\mathbf{q}}$ is a linear map, called a *structure morphism* (sometimes referred to as a *transition morphism*);
- the structure morphisms and their compositions commute, i.e. for $\mathbf{p} \preceq \mathbf{q} \preceq \mathbf{r}$, the following diagram of structure morphisms commutes:

$$\begin{array}{ccc}
& M_{\mathbf{q}} & \\
\nearrow & & \searrow \\
M_{\mathbf{p}} & \longrightarrow & M_{\mathbf{r}}
\end{array}$$

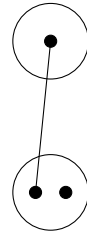
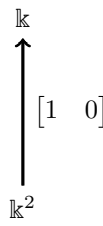
- and any structure morphism $M_{\mathbf{p}} \rightarrow M_{\mathbf{p}}$ is the identity.

When P isn't specified, M is called a *poset module*. An $x \in M_{\mathbf{p}}$ is referred to as an *element*.

Remark 2.3.6. A *basis* for a persistence module M (as a $\mathbb{k}$ -vector space) is a choice of a basis for each vector space $M_{\mathbf{p}}$.

Motivation 2.3.7 (Relevance of poset modules to persistence). The output of persistent homology, called a *persistence module*, is a poset module (definition 2.3.5).

Example 2.3.8 (Diagrams of poset modules). Poset modules are sometimes drawn as Hasse diagrams adorned with vector spaces and linear maps. The three diagrams of P -modules below right are equal. Not all poset modules can be represented by diagrams like that on the far right.



diagrams of a P -module ($\mathbb{k}$ is a field)

Definition 2.3.9. A *morphism of P -modules* is a degree-preserving linear map. More explicitly: a morphism $M \rightarrow N$ is a collection of linear maps $\{M_{\mathbf{p}} \rightarrow N_{\mathbf{p}}\}_{\mathbf{p}}$ that commute with the structure morphisms $\{M_{\mathbf{p}} \rightarrow M_{\mathbf{q}}\}_{\mathbf{p} \leq \mathbf{q}}$ and $\{N_{\mathbf{p}} \rightarrow N_{\mathbf{q}}\}_{\mathbf{p} \leq \mathbf{q}}$.

Example 2.3.10. Below are two P -modules and an isomorphism between them.

Example 2.3.11. A $\mathbb{Z}^n$ -module is a poset module but can also be described as a $\mathbb{Z}^n$ -graded module over the polynomial ring $\mathbb{k}[x_1, \dots, x_n]$.

Remark 2.3.12. A P -module M is a functor from a poset P to the category of vector spaces over a field $\mathbb{k}$. A P -module homomorphism is a natural transformation. This functorial point of view is taken in many works on multipersistence, e.g. [CSV14; BSS15; Cur15; Les15; LW15; GC17; Mil17; SC+17].

2.3.2 Persistence

Persistent homology takes, as input, a P -indexed *filtration* of a topological space (definition 2.3.13) and gives, as output, a P -module called a *persistence module* (definition 2.3.14). The poset P is called the *parameter space* and is frequently $\mathbb{Z}^n$, $\mathbb{R}^n$, or a finite poset.

The study of $P = \mathbb{N}^n$ -modules is part of combinatorial (or sometimes multigraded) commutative algebra. Many of the proof techniques for statements concerning general P -modules depend on reducing to the $P = \mathbb{N}^n$ case.

Definition 2.3.13. A P -indexed *filtration* of a topological space X is a collection of subspaces $\{X_{\mathbf{p}} \subseteq X\}_{\mathbf{p} \in P}$ such that the poset structure induced by the inclusions between subspaces is equal to the poset P .

Definition 2.3.14. The i^{th} *persistent homology* of a P -indexed filtration of X is the P -module

$$M = \bigoplus_{\mathbf{p} \in P} H_i X_{\mathbf{p}}$$

with structure morphisms induced by the inclusions of the subspaces $X_{\mathbf{p}}$. The vector space $H_i X_{\mathbf{p}}$ is the (singular) homology of the topological space $X_{\mathbf{p}}$ taken with field coefficients $\mathbb{k}$. M is frequently referred to as a *persistence module*.

A reference on homology is [Hat02].

Remark 2.3.15. Persistent homology is exactly the homology functor taking a P -indexed filtration to a P -module.

Remark 2.3.16 (One vs many parameters). The definitions given above apply to both single-parameter and multiparameter persistence. The only difference between them from this perspective is the poset used as a parameter space: single-parameter persistence always has a totally ordered parameter space because it is one-dimensional; multipersistence can have a multidimensional parameter spaces like $\mathbb{R}^n$, or any arbitrary poset for a parameter space. Persistence over arbitrary posets is sometimes called *generalized persistence*.

There is significant difference between single-parameter and multiparameter persistence in their algebraic complexity. See remark 2.3.18 for an example of how multipersistence is algebraically more complicated than single-persistence.

2.3.3 Persistence translation: diagrams/barcodes $\leftrightarrow$ modules

Some readers may be more familiar with viewing persistence through a persistence diagram. For those readers' convenience, we make a direct translation between a persistence diagram and a corresponding persistence module.

Remark 2.3.17. Persistence diagrams hold slightly less information than barcodes and persistence modules because they don't determine whether births and deaths are open or closed. We will ignore this technicality in this manuscript by assuming all points in persistence diagrams correspond to closed bars in barcodes. This is not a standard assumption (bars are frequently assumed to be open on the right) but it avoids technical details about open deaths described in [Mil17, Section 1.8].

Since persistence diagrams are restricted to single-parameter persistence, we assume there is only one parameter and the parameter space is $\mathbb{Z}$ or $\mathbb{R}$.

To translate between persistence diagram and persistence module, first translate from persistence diagram to barcode. The common language between persistence diagrams and barcodes is homology classes and their births and deaths. A homology class that is born at point $\mathbf{b}$ and dies at point $\mathbf{d}$ corresponds to

- a point $(\mathbf{b}, \mathbf{d}) \in P \times P$ in a persistence diagram, and
- a bar starting at $\mathbf{b}$ and ending at $\mathbf{d}$ in a barcode.

See the top row of figure 2.5 for a color-coded correspondence. Technically, this correspondence between homology classes and bars is only true when using a specific basis for the homology. In general homology classes are linear combinations these specific basis elements.

A barcode can be interpreted as a diagram of a persistence module. At every point $\mathbf{p}$ in the parameter space there is a vector space $M_{\mathbf{p}}$. The dimension of $M_{\mathbf{p}}$ is the number of bars above $\mathbf{p}$ in the barcode. The maps $M_{\mathbf{p}} \rightarrow M_{\mathbf{q}}$ are diagrammed by the barcode: each bar over $\mathbf{p}$ is a homology class and if a bar over $\mathbf{p}$ extends enough to also lie over $\mathbf{q}$ then the bar's homology class in $M_{\mathbf{p}}$ maps to the bar's homology class in $M_{\mathbf{q}}$. If a bar over $\mathbf{p}$ dies before reaching $\mathbf{q}$, then the corresponding homology class goes to 0 in the map $M_{\mathbf{p}} \rightarrow M_{\mathbf{q}}$. (See bottom row of figure 2.5; at left is a barcode with parameters $\mathbf{p}$ and $\mathbf{q}$ labeled and center and right are representations of the corresponding module.)

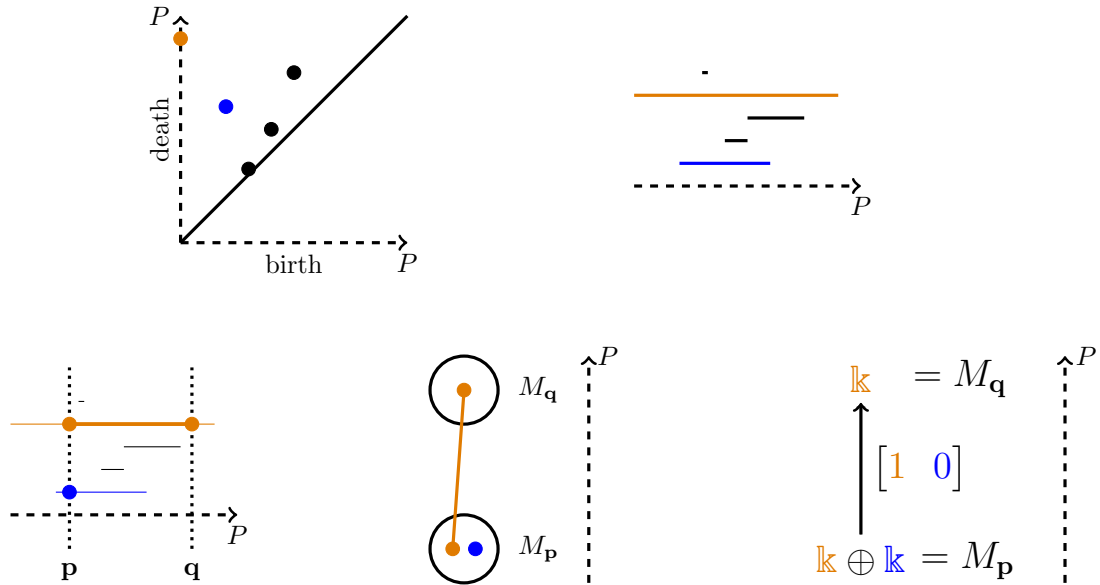


Figure 2.5. Comparing persistence diagrams, barcodes, and modules. Orange and blue each represent their own homology class. The orange homology class can be observed from the two different parameter values $\mathbf{p}$ and $\mathbf{q}$.

The bottom center and right diagrams in figure 2.5 show only part of the persistence module. Not shown are vector spaces for every other point in the parameter space and the corresponding maps.

Remark 2.3.18. Multipersistence modules are not as simply structured as single

persistence modules. Unlike in single-persistence, there is not always a basis for a multipersistence module such that structure morphisms come from set maps as above. That implies there aren't, in general, barcodes for multipersistence modules.

2.4 Tameness and poset module encodings

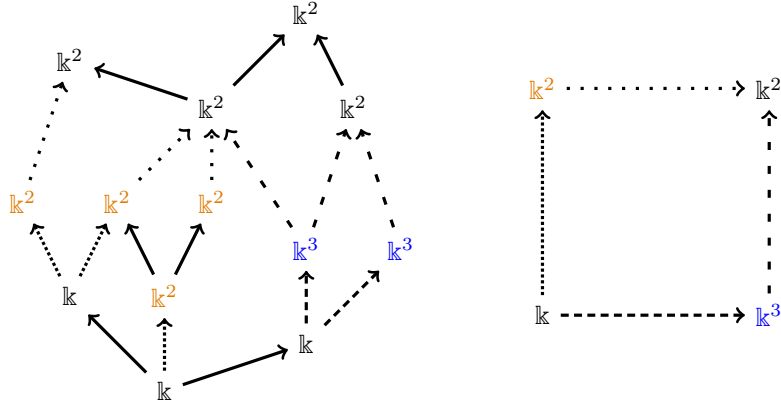
Some poset modules are difficult to work with because they are too complicated. For example an infinitely generated poset module does not necessarily admit a finite presentation and therefore might not be easy to work with theoretically or on a computer. But though multipersistence modules constructed from data can be infinitely generated, many can still be described in a finite way. See example 2.4.8 for an infinitely generated (and infinitely cogenerated, definition 3.3.2) module from the fly wing application in chapter 5. This module can be described using a finite poset module and finitely many semialgebraic sets.

This section describes which poset modules have enough structure to allow for their analysis via the invariants and metrics described in later chapters. *Tame* modules (definition 2.4.5) will be the ones that are simple enough to work with and *encodings* (definition 2.4.3) will expand the space of modules that can be considered tame to include even some infinitely generated modules.

Motivation 2.4.1 (Relevance to persistence). Parameter spaces for persistence applications are frequently $\mathbb{R}^n$, which is a very large poset, but the resulting persistence modules do not always need every point in $\mathbb{R}^n$ to describe their data. Encodings are a way to make modules over large parameter spaces into modules with the same information but living over a much smaller parameter space. Ideally the new, smaller

parameter space is easier to work with, like a finite poset or $\mathbb{Z}^n$.

Example 2.4.2. Morphisms drawn with solid arrows are the identity. The two modules below have common characteristics: the right module's vector spaces are in many-to-one correspondence with the left module's, and that correspondence respects the linear maps between vector spaces. We say that the module on the right is an *encoding* of the module on the left. Given that we remember the poset morphism used to create the encoding and we have the encoded module on the right, we are able to reconstruct the module on the left.



Definition 2.4.3 ([Mil17, Definition 2.6]). P is a poset. An *encoding* of a P -module M is a poset morphism $\pi : P \rightarrow Q$ and a Q -module H such that

$$M \cong \pi^* H = \bigoplus_{\mathbf{p} \in P} H_{\pi(\mathbf{p})}.$$

$(\pi : P \rightarrow Q, H)$ is a *finite encoding* if the poset Q is finite and the vector space $H_{\mathbf{q}}$ is finite dimensional for all $\mathbf{q} \in Q$.

An encoding is a change-of-poset for a poset module. Encodings were defined in [Mil17] to translate $\mathbb{R}^n$ -modules into $\mathbb{Z}^n$ -modules, which are well understood and can

be computed using software like RIVET ([KLW14]). Admitting a finite encoding is a necessary condition for a module to be tame. This is essentially a Noetherianity (or finitely generatedness) condition. The other finiteness condition needed for tameness is *P-finiteness*.

Definition 2.4.4. A P -module M is *P-finite* (sometimes called *pointwise finite-dimensional*) if the vector space $M_{\mathbf{p}}$ is finite dimensional for every $\mathbf{p} \in P$.

Definition 2.4.5. A P -finite module over a poset P is *tame* if it admits a finite encoding.

A relevant question when discussing encodings is “How do we construct a finite poset (or a poset morphism to $\mathbb{Z}^r$) that makes encoding worthwhile?” More details are given in [Mil17], but listed below is a basic explanation of how to construct a finite encoding and some results about when a finite encoding exists.

Definition 2.4.6 ([Mil18]). Fix a poset P and a P -module M . A *constant subdivision* of P *subordinate* to M is a partition of P into *constant regions*. Each constant region $I \subseteq P$ has associated to it a vector space M_I and isomorphisms $\{M_I \rightarrow M_{\mathbf{i}}\}_{\mathbf{i} \in I}$ such that for any constant regions I and J , all comparable points $\mathbf{i} \preceq \mathbf{j}$ with $\mathbf{i} \in I, \mathbf{j} \in J$ induce the same composition $M_I \rightarrow M_{\mathbf{i}} \rightarrow M_{\mathbf{j}} \rightarrow M_J$.

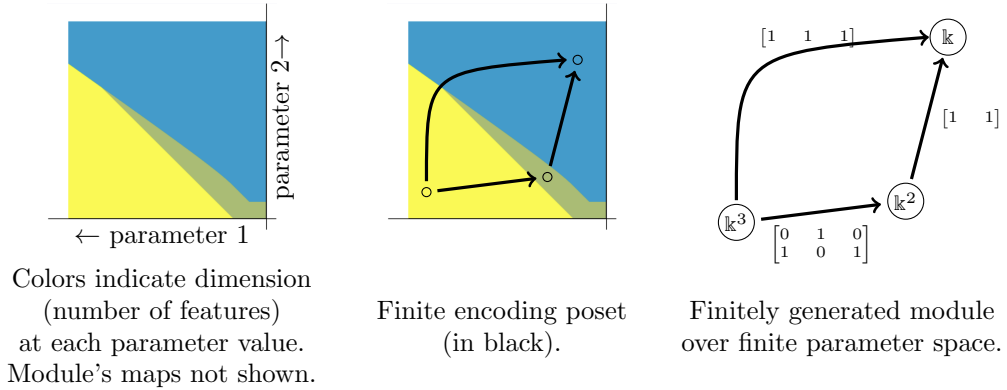
A constant subdivision is *finite* if its partition of the poset is finite.

The commutativity of compositions across different constant regions guarantees there is no monodromy in the constant subdivision. This means that the constant regions and their associated vector spaces fit neatly enough together that they can be thought of as building blocks for a smaller, simpler poset with which to encode M .

One problem with using constant regions to encode a module is that constant regions do not always form a poset. We can use *uptight regions* to refine a constant subdivision into a new constant subdivision whose constant regions form a poset [Mil17]. If the original constant subdivision is finite then the refined one constructed using uptight regions is also finite. The result of this construction is the following theorem:

Theorem 2.4.7 ([Mil18]). *Fix a P -finite module M over a poset P . M admits a finite encoding if and only if there exists a finite constant subdivision of P subordinate to M .*

Example 2.4.8. Below is an $\mathbb{R}^2$ module with three constant regions that can be finitely encoded by a three point poset. This particular example is from the wing venation application and can be found in subsection 5.4.1. This was the example that led to the author realizing that multipersistence modules from data can be infinitely generated.



Remark 2.4.9. Many of the results in chapter 3 are for modules over finite posets. At first glance requiring a finite parameter space seems restrictive, but encodings allow results over finite posets to be translated into results over tame modules. The details of this translation are the subject of future work.

Chapter 3

Invariants and summary statistics

Persistence modules aren't amenable to statistics so properties of those modules are measured and used as summary statistics instead. A module property can only be a useful summary statistic for persistence modules if it is an isomorphism invariant, else it would change with a change of basis and might not be consistent across different implementations of the same algorithms. So algebraic invariants are used as summary statistics for persistence modules, which is why the two terms are used interchangeably.

There are many persistence invariants, some of which only apply to single-persistence. This chapter gives an overview of existing invariants and then introduces new invariants: the multirank function (section 3.2) and some geometric invariants that can be computed from multirank (section 3.3).

3.1 Known invariants

Remark 3.1.1 (Strictly single-persistence invariants). The barcode and persistence diagram invariants are only used in single-persistence because multipersistence modules don't have the simple algebraic structure needed to construct them.

There is a multipersistence analogue to the barcode but because multipersistence modules do not have a distinguished basis the way single-persistence modules do, it is not

an invariant. The exception is for zeroth persistent homology, which has a distinguished basis given by the connected components. Examples of “zeroth multipersistence barcodes” are the diagrams of modules in example 3.2.9.

The rest of the invariants will be described for multipersistence.

Remark 3.1.2 (A list of multipersistence invariants).

- 1) Presentations and resolutions of $\mathbb{Z}^n$ -modules can be computed [CSZ09; CSV14; LW15], but they are too unwieldy to fit into an existing statistical framework. The problem lies partially in the linear maps; their size (as matrices) is determined by the number of features (squared). In the past these maps have been substituted with their ranks (i.e., the rank function) for statistical analysis.
- 2) The *rank function* invariant (definition 3.1.4) maps a pair of degrees to either the rank of the map between them or, if such a map doesn’t exist, 0. Rank does not give information about pairs of incomparable points. In addition, rank cannot determine when a new homology classes is generated (born) or goes to 0 (dies) in multipersistence modules.
- 3) The *Hilbert function* (definition 3.1.3) holds strictly less information than the rank function: for every point in the parameter space the Hilbert function returns the dimension of the corresponding vector space.
- 4) The Hilbert polynomial and Hilbert series techniques of [HO+17] do not give any more information than the rank function.
- 5) The associated prime and local cohomology techniques of [HO+17] give the directions in which features “live forever” and a notion of how long they live in other directions, but cannot keep track of exactly where those features appear or disappear. We use primary decomposition to modify existing metrics in

section 4.3.

- 6) The *higher Betti number* invariants, suggested in [CZ09] and studied in [Knu07], indicate the number of generators, relations, and higher relations (syzygies) at each parameter value. They also determine births and deaths: $\beta_0(\mathbf{p})$ is the number of births at parameter value $\mathbf{p}$ in a $\mathbb{Z}^n$ -module and $\beta_n(\mathbf{p})$ is the number of deaths. These invariants only give local information and do not necessarily detect which generators correspond to which relations. Without being able to relate the existence of one generator or relation to another, they cannot necessarily determine a link between any pair of features. Careful: these Betti numbers are not the dimensions of the homology vector spaces; they are dimensions of the modules in a minimal free resolution of the persistence module. More specifically: $\beta_i(\mathbf{p})$ is the dimensions of the $\mathbf{p}^{th}$ graded piece of the homological degree i module in a minimal free resolution of a persistence module.

The Hilbert and rank functions (invariants 2 and 3) can be used to define metrics on persistence modules and are frequently used for data analysis. Since they will be referred to later, included below are exact definitions of both.

Definition 3.1.3. The *Hilbert function* $\text{Hilb}_M : P \rightarrow \mathbb{N}$ of a P -module M maps a point in the parameter space to the dimension of its corresponding vector space:

$$\text{Hilb}_M : \mathbf{p} \mapsto \dim_{\mathbb{k}} M_{\mathbf{p}}$$

Definition 3.1.4. The *rank function* $\text{rank}_M : P \times P \rightarrow \mathbb{N}$ of a P -module M takes an (ordered) pair of points in the parameter space to the rank of the structure morphism

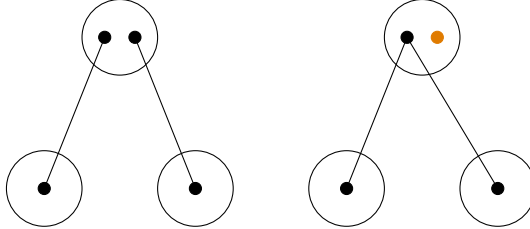
from the first to the second:

$$\text{rank}_M : (\mathbf{p}, \mathbf{q}) \mapsto \text{rank}(M_{\mathbf{p}} \rightarrow M_{\mathbf{q}})$$

The rank function on the right is the rank of a linear map. If $\mathbf{p} \not\leq \mathbf{q}$ then the rank function returns 0.

Most of the computationally viable invariants listed above do not give any more information than the rank invariant (the exceptions are the higher Betti numbers and primary decompositions). The following example, which has also been presented in [LW15; HO+17], shows how the rank invariant is not able to capture births (and dually deaths) in a multipersistence module.

Example 3.1.5. The following modules are non-isomorphic but have the same rank invariant. The module on the right has a birth, marked in orange, at the maximal point in the poset. The module on the left has no birth at the maximal point.

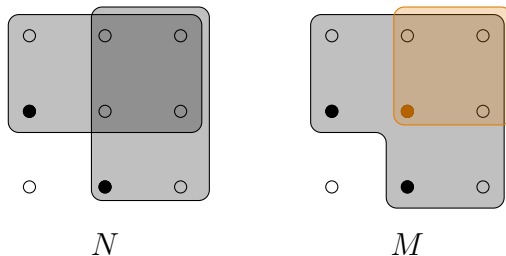


In [LW15] and [HO+17] this same example is given in terms of generators and relations for $\mathbb{Z}^2$ -modules. Those presentations and diagrams of the resulting modules are provided below. In this setting we can also see that the modules have the same

associated primes (just the $\langle 0 \rangle$ ideal).

$$N = S(-1, 0) \oplus S(0, -1)$$

$$M = \frac{S(-1, 0) \oplus S(0, -1) \oplus S(-1, -1)}{\text{im}(x_2, -x_1, 0)^t}$$



In the next section, the *multirank invariant* (definition 3.2.4) will detect that the modules in example 3.1.5 are non-isomorphic and will be sufficient to compute the births (and deaths) of each module.

3.2 The multirank invariant

A summary statistic that is a perfect descriptor for single-parameter persistence modules is the *rank function invariant* (definition 3.1.4). In single-persistence, the rank function is a *complete invariant*, which means it fully determines a persistence module. The rank function can also be used to compute other invariants that describe specific geometric features associated to a persistence module, namely the births, deaths, and persistence of homology classes. These auxiliary invariants that give geometric interpretations of data are particularly useful for understanding applications.

The rank function does not give complete information in the multiparameter persistence setting. It is able to determine persistence of homology classes, but it does not determine births and deaths. An extension of the rank function, called the *multirank function* (definition 3.2.4), also fails to be complete (see example 3.2.11) but does provide enough information to determine some auxiliary invariants with useful geometric meaning, including births and deaths.

The limitation of the rank function for multipersistence modules is that rank does not detect relations between homology classes. In single-persistence there is always a basis for homology such that there are no nontrivial relations among distinct basis elements. With no relations between classes, the rank function’s inability to keep track of basis element interactions is unimportant. But in multipersistence, there is not necessarily such a neat basis. The multirank invariant is introduced below (definition 3.2.4) as a fix for this deficiency in the rank function.

The following work will be available in [Tho19].

Assumption 3.2.1. For the rest of this chapter parameter spaces will be finite posets unless otherwise stated. See remark 2.4.9 for reasoning why this assumption is not particularly restrictive.

3.2.1 Definition

Notation. Subsets of a poset are usually capital (e.g. $A \subseteq P$). Degrees are lowercase bold (e.g. $\mathbf{a} \in P$) to suggest their relation to vectors or tuples in $\mathbb{N}^n$, $\mathbb{Z}^n$, or $\mathbb{R}^n$. Subsets of a poset notated by S and T are chosen to suggest that they correspond to the “source” and “target” of a morphism.

Definition 3.2.2. Suppose P is a poset, S and T are subset of P , and $\mathbf{s}, \mathbf{t} \in P$.

- 1) The *upset* generated by S is the subposet

$$P_{\succeq S} = \{\mathbf{p} \in P : \mathbf{p} \succeq \mathbf{s} \text{ for some } \mathbf{s} \in S\}$$

with the inherited partial order.

- 2) The *open upset* generated by a point $\mathbf{s} \in P$ is the subposet

$$P_{\succ \mathbf{s}} = \{\mathbf{p} \in P \mid \mathbf{p} \succ \mathbf{s}\}.$$

Recall that $\succ$ is a strict inequality, i.e. $\mathbf{p} \neq \mathbf{s}$.

- 3) The *downset* generated by T is the subposet

$$P_{\preceq T} = \{\mathbf{p} \in P \mid \mathbf{p} \preceq \mathbf{t} \text{ for some } \mathbf{t} \in T\}.$$

- 4) The *open downset* generated by a point $\mathbf{t} \in P$ is the subposet

$$P_{\prec \mathbf{t}} = \{\mathbf{p} \in P \mid \mathbf{p} \prec \mathbf{t}\}.$$

Definition 3.2.3. Suppose S and T are subsets of a poset P . S *precedes* T ($S \preceq T$) if $\mathbf{s} \preceq \mathbf{t}$ for every $\mathbf{s} \in S$ and $\mathbf{t} \in T$.

Definition 3.2.4. For a P -finite module M over a finite poset P and subsets $S, T \subseteq P$, the *multirank* from S to T is

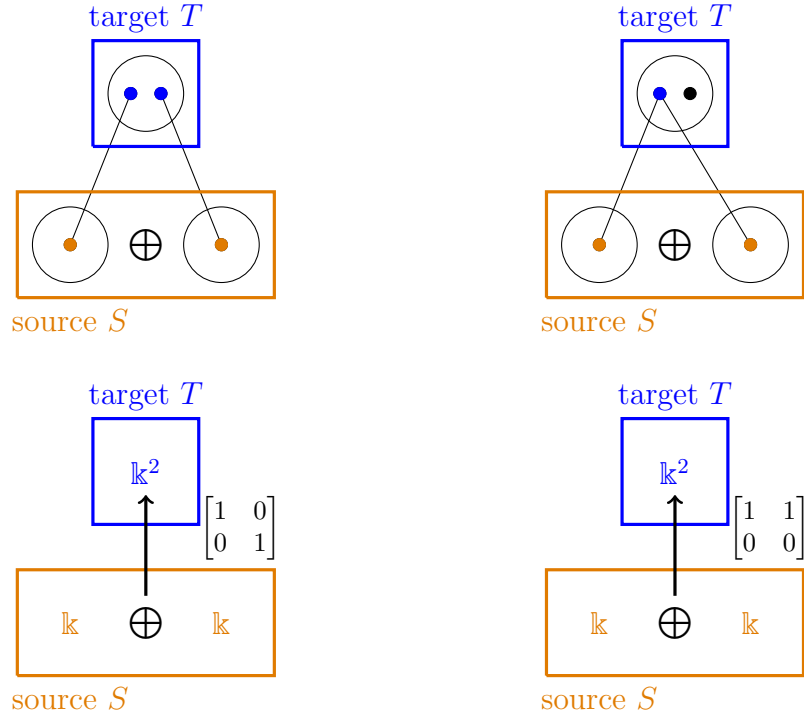
$$\text{multirank}_M(S, T) = \text{rank} \left(\bigoplus_{\mathbf{s} \in S} M_{\mathbf{s}} \rightarrow \bigoplus_{\mathbf{t} \in T} M_{\mathbf{t}} \right)$$

where

$$M_{\mathbf{s}} \rightarrow M_{\mathbf{t}} = \begin{cases} \text{structure morphism of } M & \mathbf{s} \preceq \mathbf{t} \\ 0 & \mathbf{s} \not\preceq \mathbf{t} \end{cases}$$

We call S the *source set* and T the *target set*.

Example 3.2.5. Let's revisit example 3.1.5 (the modules are reproduced below). Compute multirank with the minimal points as the source and the maximal point as the target. We can see multirank can distinguish between these two non-isomorphic modules even when the rank function cannot.



$$\begin{aligned} \text{multirank}_{(S, T)} &= \text{rank} \left(\bigoplus_{\mathbf{s} \in S} M_{\mathbf{s}} \rightarrow \bigoplus_{\mathbf{t} \in T} M_{\mathbf{t}} \right) \\ &= \text{rank} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 2 \qquad \qquad \qquad = \text{rank} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = 1 \end{aligned}$$

3.2.2 Properties

A measure of a persistence module is only useful if it does not depend on a basis for the module, i.e., it is an isomorphism invariant. If it's not an invariant then measurement of the property may be inconsistent, either because of using different algorithms or different bases. The multirank function is indeed an invariant:

Theorem 3.2.6. *The multirank function of a poset module is invariant under poset module isomorphism.*

Proof. Ranks of linear maps are invariant under vector space isomorphism. □

Proposition 3.2.7. *The multirank invariant contains strictly more information than the Hilbert and rank function invariants.*

Proof. When the source and target sets are singletons $S = \{\mathbf{s}\}$ and $T = \{\mathbf{t}\}$, $\text{multirank}(S, T)$ returns the same value as $\text{rank}(M_{\mathbf{s}} \rightarrow M_{\mathbf{t}})$. The rank function already contains all of the information in the Hilbert function via $\text{Hilb}_M(\mathbf{p}) = \text{rank}(M_{\mathbf{p}} \rightarrow M_{\mathbf{p}})$. Example 3.2.5 shows that the multirank contains information that the rank invariant does not. □

Reducing the size of the source and target sets for a multirank computation reduces the size of the matrix whose rank must be computed (see remark 3.4.4). Fortunately, the source and target sets used to compute a multirank can sometimes be pruned:

Lemma 3.2.8. *If $S \preceq T$, the multirank from S to T depends only on the downset generated by S and the upset generated by T .*

Proof. Compare $\text{multirank}(S, T)$ to $\text{multirank}(P_{\preceq S}, T)$ for $S \preceq T \subseteq P$. For every $\mathbf{p}$

in the downset generated by S , the map $M_{\mathbf{p}} \rightarrow \bigoplus_{\mathbf{t} \in T} M_{\mathbf{t}}$ factors through the map $\bigoplus_{\mathbf{s} \in S} M_{\mathbf{s}} \rightarrow \bigoplus_{\mathbf{t} \in T} M_{\mathbf{t}}$. So replacing S by $P_{\preceq S}$ in $\bigoplus_{\mathbf{s} \in S} M_{\mathbf{s}} \rightarrow \bigoplus_{\mathbf{t} \in T} M_{\mathbf{t}}$ does not change the map's image and therefore does not change the map's rank, either.

The proof that $\text{multirank}(S, T) = \text{multirank}(S, P_{\succeq T})$ is similar. $\square$

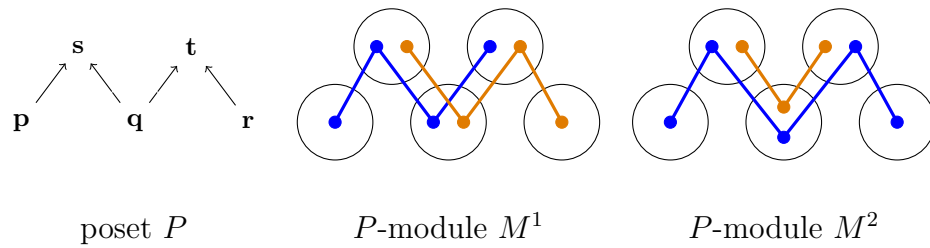
Many intuitive properties of multirank only hold when the source and target sets satisfy $S \preceq T$. But if we decided to only consider the multirank function for pairs (S, T) with $S \preceq T$, we would lose some information:

Example 3.2.9. The modules M^1 and M^2 below have all of the same multiranks from S to T when $S \preceq T$.

For example, for both modules:

$$\text{multirank}_{M^i}(\{\mathbf{p}\}, \{\mathbf{s}\}) = \text{rank}(M_{\mathbf{p}}^i \rightarrow M_{\mathbf{s}}^i) = 1$$

$$\text{multirank}_{M^i}(\{\mathbf{p}, \mathbf{q}\}, \{\mathbf{s}\}) = \text{rank}(M_{\mathbf{p}}^i \oplus M_{\mathbf{q}}^i \rightarrow M_{\mathbf{s}}^i) = 2$$



But, multirank distinguishes M^1 and M^2 when $S = \{\mathbf{p}, \mathbf{q}, \mathbf{r}\} \not\preceq T = \{\mathbf{s}, \mathbf{t}\}$:

$$\text{multirank}_{M^1}(\{\mathbf{p}, \mathbf{q}, \mathbf{r}\}, \{\mathbf{s}, \mathbf{t}\}) = 4$$

$$\text{multirank}_{M^2}(\{\mathbf{p}, \mathbf{q}, \mathbf{r}\}, \{\mathbf{s}, \mathbf{t}\}) = 3$$

Remark 3.2.10. From the above example we can conclude that multirank is not a complete invariant on tame poset modules if it is only defined for subsets $S \preceq T$ in the poset P .

In fact, even without restructuring the domain, multirank is not a complete invariant for the space of tame poset modules:

Example 3.2.11. This example was inspired by [DW05], which is an introductory reference for quiver representations and Gabriel’s theorem.

We begin by trying to construct an isomorphism (dotted arrows) between the two poset modules below. The only way to do so is to make the map multiplication by a scalar. But, if $\lambda \neq \mu$, then there is no isomorphism between the modules.

Furthermore, if $\lambda, \mu \notin \{0, 1\}$, then within each module every pair of structure morphisms is linearly independent. So the multirank functions for the two modules are the same: the Hilbert functions coincide, the rank functions coincide, and every nontrivial higher multirank is 2.

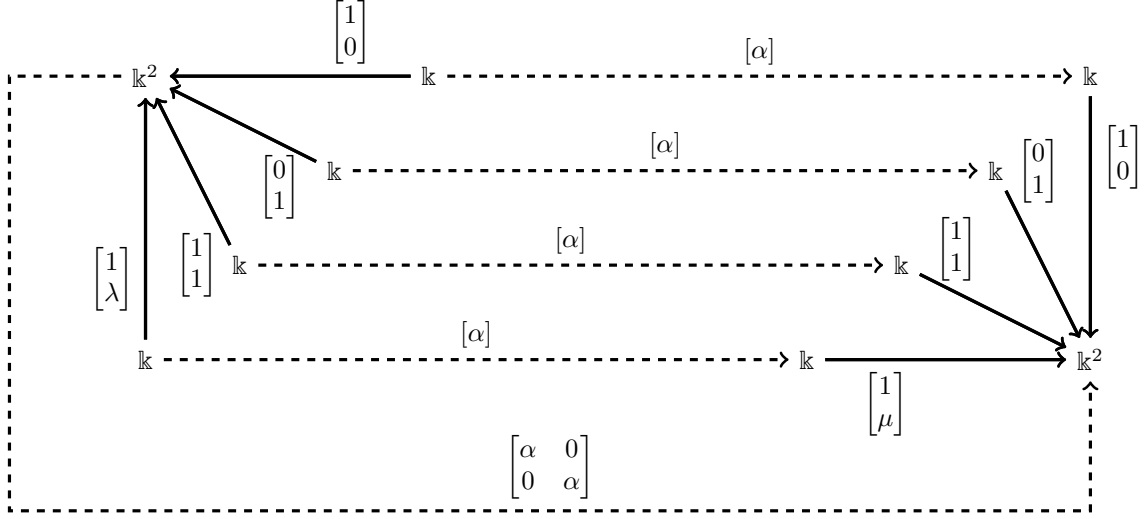


Figure 3.5. Two P -modules that have the same multirank but are non-isomorphic when $\lambda \neq \mu$ and $\lambda, \mu \notin \{0, 1\}$.

We end this subsection with a conjecture on the completeness of the multirank invariant.

Conjecture 3.2.12. *Multirank is a complete invariant for tame persistent 0^{th} persistence modules.*

What suggests that conjecture 3.2.12 might be true? A finite encoding of a tame module M determines M up to isomorphism by definition 2.4.3, so multirank is a complete invariant for tame 0^{th} persistence modules if it is a complete invariant for their finite encodings.

What suggests that conjecture 3.2.12 might be true for P -finite 0^{th} persistence modules over finite posets P ? Multirank is a combinatorial object but since the modules in question are combinatorial objects as well, multirank has a chance of fully describing them. Indeed, finite encodings of 0^{th} persistence modules are combinatorial because they can be induced by *set systems* corresponding to connected components of the

persistence filtration. Set systems are like poset modules but the vector spaces are replaced by sets, the structure morphisms are set maps, and the isomorphisms between set systems are bijective set maps.

Listed below are some examples related to the completeness of multirank for tame 0^{th} persistence modules.

- 1) The counterexample to the completeness of multirank for poset modules in example 3.2.11 relies on the existence of four pairwise-linearly-independent structure morphisms $\mathbb{k} \rightarrow \mathbb{k}^2$. There are only two set maps $\mathbb{k} \rightarrow \mathbb{k}^2$, so we cannot construct four structure morphisms that are both pairwise-linearly-independent and induced by set maps. So, example 3.2.11 cannot be modified to be a counterexample to conjecture 3.2.12.
- 2) The fact that multirank can't be complete without taking into account source and target sets $S \not\subseteq T$ (see example 3.2.9) disrupts the intuition described in subsection 3.2.3.
- 3) Set systems are not invariants of P -finite 0^{th} persistence modules over finite P . Said another way: two non-isomorphic set systems can induce isomorphic modules.

3.2.3 Intuition

What does multirank measure? Multirank is an extension of rank in that when the source and target sets are singletons, multirank and rank coincide. Rank measures the number of independent homology classes alive across a given interval, and this holds as long as the starting parameter value precedes the ending parameter value.

A multirank with source set preceding its target set also measures the number of independent homology classes alive across an “interval;” this interval just happens to be “wider.”

Why does multirank avoid overcounting the number of homology classes? The direct sums of the source (resp. target) vector spaces cause linear dependences between elements that come from different points in the parameter space but represent the same homology class. Multirank only counts linearly independent homology classes, so it avoids double-counting. This property does not hold when the source set fails to precede the target set. In that case some of the linear relations (which come from the structure morphisms) are missing so classes that should be identified end up each contributing to the multirank.

Example 3.2.13. Consider a persistence module M that is a free $\mathbb{Z}^n$ -module generated by a vector space of dimension one at the origin (i.e., at all points at or above the origin there is also a vector space of dimension one, and between vector spaces at comparable points there are identity maps). This module has one independent homology class that is born at the origin and never dies.

Suppose $\mathbf{p} \preceq \mathbf{q}$ are points in $\mathbb{N}^n$. Then

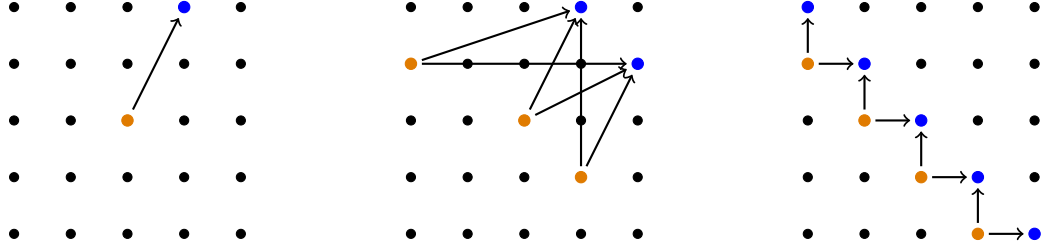
$$\text{multirank}(\mathbf{p}, \mathbf{q}) = \text{rank}(M_{\mathbf{p}} \rightarrow M_{\mathbf{q}}) = 1$$

Suppose $\{\mathbf{p}_1, \dots, \mathbf{p}_r\} \preceq \{\mathbf{q}_1, \dots, \mathbf{q}_s\}$ are two subsets of $\mathbb{N}^n$. Then

$$\text{multirank}(\{\mathbf{p}_1, \dots, \mathbf{p}_r\}, \{\mathbf{q}_1, \dots, \mathbf{q}_s\}) = 1$$

These two multirank computations agree with idea that though there are many vector

space elements (all at different points in the parameter space $\mathbb{Z}^n$), there is really only one independent homology class.



If we choose a source set along an antidiagonal and a target set along a different antidiagonal, like

$$S = \{(3, 0), (2, 1), (1, 2), (0, 3)\}$$

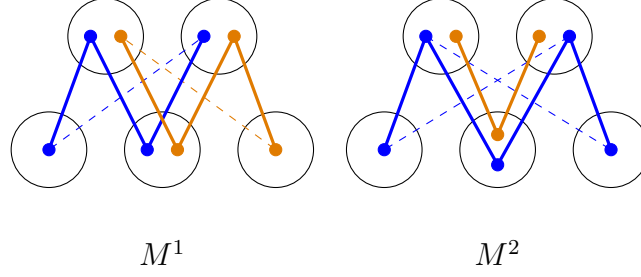
$$T = \{(4, 0), (3, 1), (2, 2), (1, 3), (0, 4)\}$$

Then we'll get a multirank that doesn't describe our free rank 1 module very well:

$$\text{multirank}(\textcolor{brown}{S}, \textcolor{blue}{T}) = 4$$

Example 3.2.14. In both modules of example 3.2.9, there are only two distinct homology classes but the multirank from the minimal points to the maximal points still attains the value 4 on M^1 and 3 on M^2 . If we could compute the “missing” structure morphisms to make all the minimal points (**p**, **q**, and **r**) precede all the maximal points (**s** and **t**), then for both modules

$$\text{multirank}_{M^i}(\{\mathbf{p}, \mathbf{q}, \mathbf{r}\}, \{\mathbf{s}, \mathbf{t}\}) = 2$$



3.3 Geometric invariants from multirank

Multirank holds combinatorial information about the structure maps of a persistence module, but not all of its information is interpretable. It would be nice to have interpretable, geometric invariants that translate directly to applications; ie invariants that directly measure interesting properties, or *features* of the homology.

Here are three types of behavior of homology classes that are “flowing” through a persistence module. They can all be computed from multirank.

- 1) Appearance and disappearance features: Where in the parameter space do births and deaths of features occur?
- 2) Merging and branching of features: How do features born at different points in the poset interact when they flow into the same point? Are these inherently the same homology class just observed at different points in the parameter space? Are they distinct features that later merge (i.e., map into the same feature)? Or do they never interact meaningfully with one another?
- 3) Persistence of features: how many features’ lives span a given interval of parameter values?

Invariant 3 was described in subsection 3.2.3 and is the geometric interpretation of the multirank function. The other two invariants will be explored via invariants that explicitly describe these geometric characteristics and can be computed from the multirank invariant.

Remark 3.3.1. For all three of the invariants listed, the geometric interpretation only applies on a restriction of the domain of the multirank function. Their geometric descriptions only apply when $S \preceq T$ in the same way that invariant 3 only applies when $S \preceq T$ (see subsection 3.2.3).

3.3.1 Births and deaths

For a P -module M , a *birth* at $\mathbf{p}$ is an element of $M_{\mathbf{p}}$ that is outside the image of every structure morphism of M mapping into $M_{\mathbf{p}}$. A *death* at $\mathbf{p}$ is an element that, pushed into any poset degree strictly above $\mathbf{p}$, becomes 0.

Births and deaths are topological ideas. In the setting of modules over finite posets, their algebraic counterparts are generators and cogenerators, respectively. (This is also the case for all discrete posets, but in continuous posets there are the complications of *open* and *closed* generators and cogenerators. See [Mil17] for details.)

Definition 3.3.2. For a finite poset P and tame P -module M , the *top* and *socle* at a degree $\mathbf{p} \in P$ are the vector spaces

$$(\text{top } M)_{\mathbf{p}} = \text{coker} \left(\bigoplus_{\mathbf{q} < \mathbf{p}} M_{\mathbf{q}} \rightarrow M_{\mathbf{p}} \right)$$

$$(\text{soc } M)_{\mathbf{p}} = \ker \left(M_{\mathbf{p}} \rightarrow \bigoplus_{\mathbf{q} > \mathbf{p}} M_{\mathbf{q}} \right)$$

A *generator* of degree $\mathbf{p}$ is a nonzero element of $(\text{top } M)_{\mathbf{p}}$ and a *cogenerator* of degree $\mathbf{p}$ is a nonzero element of $(\text{soc } M)_{\mathbf{p}}$.

Remark 3.3.3. Generators (births) and cogenerators (deaths) represent segments of the lower and upper boundaries, respectively, for the lives of homology classes. They are dual notions (via Matlis duality: see example 3.3.4 and [MS05, Definition 11.15]) and their symmetry is only broken by the directedness of the poset: homology classes flow from births to deaths.

Example 3.3.4. Suppose M is a tame finite length $\mathbb{Z}^n$ -module, i.e. a module over the polynomial ring $\mathbb{k}[x_1, \dots, x_n]$ with finitely many non-zero vector spaces, each of which is finite-dimensional.

To take the Matlist dual of M , we “flip” the module (or reverse the direction of the arrows) and take vector space duals of each vector space. So in our case, the Matlis dual of M is $M^\vee$ with

$$M_{\mathbf{p}}^\vee = M_{-\mathbf{p}}$$

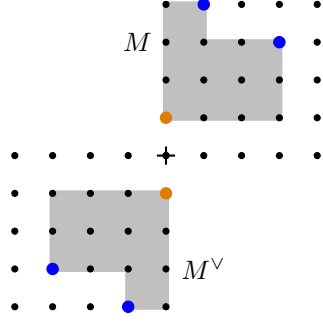
The structure morphisms of $M^\vee$ are the transposes of the structure morphisms of M .

Let $\mathfrak{m} = \langle x_1, \dots, x_n \rangle$ be the maximal ideal. The vector space $\text{top } M = \bigoplus_{\mathbf{p} \in P} (\text{top } M)_{\mathbf{p}}$ contains the elements of M that are not in the image of $\mathfrak{m}$, that is, $\text{top } M = M/\mathfrak{m}M$.

The vector space $\text{soc } M = \bigoplus_{\mathbf{p} \in P} (\text{soc } M)_{\mathbf{p}}$ contains the elements that are annihilated by $\mathfrak{m}$, i.e. the elements that go to 0 when acted on by any and all elements of $\mathfrak{m}$.

When taking the Matlis dual, the top and socle “switch.”

$$\text{top } M \longleftrightarrow \text{soc } M^\vee \qquad \text{soc } M \longleftrightarrow \text{top } M^\vee$$



Remark 3.3.5 (On (non-)canonicity of births). The following is a short exact sequence of vector spaces

$$0 \rightarrow \operatorname{im} \left(\bigoplus_{\mathbf{q} \prec \mathbf{p}} M_{\mathbf{q}} \rightarrow M_{\mathbf{p}} \right) \rightarrow M_{\mathbf{p}} \rightarrow \operatorname{coker} \left(\bigoplus_{\mathbf{q} \prec \mathbf{p}} M_{\mathbf{q}} \rightarrow M_{\mathbf{p}} \right) \rightarrow 0$$

If we set

$$M_{\prec \mathbf{p}} = \operatorname{im} \left(\bigoplus_{\mathbf{q} \prec \mathbf{p}} M_{\mathbf{q}} \rightarrow M_{\mathbf{p}} \right) \subseteq M_{\mathbf{p}}$$

then the short exact sequence splits to give

$$M_{\mathbf{p}} \cong (\operatorname{top} M)_{\mathbf{p}} \oplus M_{\prec \mathbf{p}}$$

This is a direct sum decomposition of $M_{\mathbf{p}}$ into its births and its elements that were born strictly before.

Since $(\operatorname{top} M)_{\mathbf{p}}$ is a direct summand, it is orthogonal to $M_{\prec \mathbf{p}}$. This is why births are canonical exactly when they are minimal (i.e., $M_{\prec \mathbf{p}} = 0$): Given an element $b \in (\operatorname{top} M)_{\mathbf{p}}$ and $v \in M_{\prec \mathbf{p}}$, both b and $b + v$ are representatives for the same birth.

Theorem 3.3.6. *For a finite poset P and a tame P -module M , the number of births and deaths at each parameter value can be computed from the multirank function as*

follows:

$$\#\{\text{births at } \mathbf{p}\} = \text{multirank}_M(\mathbf{p}, \mathbf{p}) - \text{multirank}(P_{\prec \mathbf{p}}, \{\mathbf{p}\})$$

$$\#\{\text{deaths at } \mathbf{p}\} = \text{multirank}_M(\mathbf{p}, \mathbf{p}) - \text{multirank}(\{\mathbf{p}\}, P_{\succ \mathbf{p}})$$

Proof. Notice that $\text{multirank}_M(\mathbf{p}, \mathbf{p}) = \dim_{\mathbb{k}} M_{\mathbf{p}}$. Then each expression follows from applying the rank-nullity theorem to the definitions of top and socle (definition 3.3.2) above. $\square$

The number of births and deaths at poset coordinate $\mathbf{p}$ are sometimes referred to as $\text{Betti}_0(\mathbf{p})$ and $\text{Bass}_0(\mathbf{p})$, respectively. See invariant 6 and [Knu07] for more on Betti numbers. Careful: these Betti numbers are not the dimensions of the homology vector spaces; they are the dimensions of modules in a minimal free resolution of the persistence module.

3.3.2 Merge and branch numbers

Multirank determines how homology classes merge as they are mapped through the parameter space but it doesn't do so in an "obvious" way. Merge and branch numbers present this geometric information in an explicit, intuitive way.

Definition 3.3.7. Suppose M is a tame P -module for finite poset P , S and T are subsets of P , and $\mathbf{s}$ and $\mathbf{t}$ are points in P .

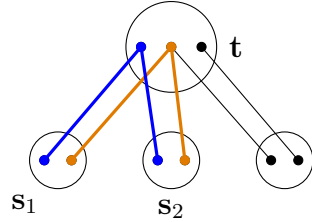
- The *merge number* $\text{merge}(S, \mathbf{t})$ is the dimension of the maximal subspace of $M_{\mathbf{t}}$ that is mapped onto surjectively by all of the $\{M_{\mathbf{s}}\}_{\mathbf{s} \in S}$.
- The *branch number* $\text{branch}(\mathbf{s}, T)$ is the dimension of the maximal subspace of

M_s that maps injectively into each of the $\{M_t\}_{t \in T}$.

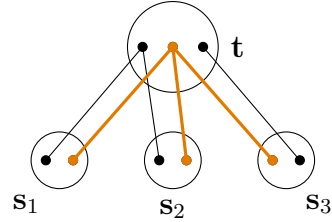
Merge and branch numbers' names were chosen to imply the geometric properties they measure when their sources precede their targets. Under that condition, a merge number counts the numbers of features in its target that come from features in all of its source points.

Example 3.3.8. Below left is a module where M_t has two distinct features that each have representatives in both M_{s_1} and M_{s_2} . The blue representatives in M_{s_1} and M_{s_2} merge as they map into M_t and are distinct from the orange representatives that also merge. The merge number is 2 since there are two distinct homology classes that have this merging property.

Below right M_t only has one feature that is represented in each of M_{s_1} , M_{s_2} , and M_{s_3} . This is a 3-way merge from M_{s_1} , M_{s_2} , and M_{s_3} into M_t .



$$\text{merge}(\{s_1, s_2\}, t) = 2$$



$$\text{merge}(\{s_1, s_2, s_3\}, t) = 1$$

Branch numbers are the dual to merge numbers: $\text{branch}(s, T)$ counts the number of features in M_s that map non-trivially into each of the $\{M_t\}_{t \in T}$. To have a branch number of greater than 1, each counted feature's forward-extending branches must be linearly independent from the others'.

The following statements relate merge and branch numbers to the multirank function

and therefore also to births and deaths.

Theorem 3.3.9. *Suppose P is a finite poset, M is a tame P -module, $S \preceq \{\mathbf{t}\}$, $\{\mathbf{s}\} \preceq T$, and $r = |S| = |T|$.*

1) *The number of r -way merges in $M_{\mathbf{t}}$ between homology classes from $\{M_{\mathbf{s}}\}_{\mathbf{s} \in S}$ is:*

$$\text{merge}(S, \mathbf{t}) = \sum_{R \subseteq S} (-1)^{|R|+1} \text{multirank}(R, \mathbf{t})$$

2) *The number of r -way branches from $M_{\mathbf{s}}$ to all of the $\{M_{\mathbf{t}}\}_{\mathbf{t} \in T}$ is*

$$\text{branch}(\mathbf{s}, T) = \sum_{R \subseteq T} (-1)^{|R|+1} \text{multirank}(\mathbf{s}, R)$$

3) *The multirank from S to $\mathbf{t}$ can be computed in terms of merge numbers:*

$$\text{multirank}(S, \{\mathbf{t}\}) = \sum_{R \subseteq S} (-1)^{|R|+1} \text{merge}(R, \mathbf{t})$$

4) *The multirank from $\mathbf{s}$ to T can be computed in terms of branch numbers:*

$$\text{multirank}(\mathbf{s}, T) = \sum_{R \subseteq T} (-1)^{|R|+1} \text{branch}(\mathbf{s}, R)$$

Proof. Statement 3: $\text{multirank}(S, \mathbf{t})$ is the dimension of the subspace of $M_{\mathbf{t}}$ spanned by each of the images of the $\{M_{\mathbf{s}} \rightarrow M_{\mathbf{t}}\}_{\mathbf{s} \in S}$. The merge numbers tell us how those images are overlapping. Statement 3 is exactly the principle of inclusion-exclusion for multirank and merge numbers.

Statement 1: This is the Möbius inversion formula applied to statement 3. For (finite)

sets R and S , the Möbius function is

$$\mu(R, S) = \begin{cases} 0 & R \not\subseteq S \\ (-1)^{|S|-|R|} & R \subseteq S \end{cases}$$

and the Möbius inversion formula is

$$g(S) = \sum_{R \subseteq S} f(R) \quad \Longleftrightarrow \quad f(S) = \sum_{R \subseteq S} g(R) \mu(R, S)$$

Apply the Möbius inversion formula for

$$g(S) = \text{multirank}(S, \mathbf{t}) \quad f(S) = (-1)^{|S|+1} \text{merge}(S, \mathbf{t})$$

Then

$$f(S) = (-1)^{|S|+1} \text{merge}(S, \mathbf{t}) = \sum_{R \subseteq S} \text{multirank}(R, \mathbf{t}) (-1)^{|S|-|R|}$$

$$\begin{aligned} \text{merge}(S, \mathbf{t}) &= \sum_{R \subseteq S} (-1)^{|S|-|R|+|S|+1} \text{multirank}(R, \mathbf{t}) \\ &= \sum_{R \subseteq S} (-1)^{|R|+1} \text{multirank}(R, \mathbf{t}) \end{aligned}$$

A reference on inclusion-exclusion and Möbius inversion is [Aig07, Sections 5.1, 5.2].

Statements 2 and 4: Similar arguments holds for branch numbers. $\square$

Corollary 3.3.10. *The parameter values and multiplicities of births and deaths of a tame P -module M over finite P can be computed from merge and branch numbers,*

respectively.

Proof. Start with theorem 3.3.6, which gives an expression for the number of births at a parameter value in terms of multirank. Theorem 3.3.9 (3) gives a formula for how to convert those multirank computations into merge number computations.

Similarly, to compute deaths in terms of branch numbers apply theorem 3.3.9 (4) to the expression for the number of deaths in theorem 3.3.6. $\square$

3.4 Computation

Below is a modification to multirank that has reduced computational cost called the *m-rank*. Following the definition of *m-rank* are examples of how to use *m-rank* and merge and branch numbers to study lifespans of features in a persistence module.

3.4.1 *m-rank*: a pruned multirank

The domain of the multirank function is the square of the power set of the poset P , which is too large for practice. For a less storage-intensive invariant, restrict the domain of multirank to $P^m \times P^m$ for some m . Call this restricted multirank the *m-rank*.

Definition 3.4.1. The *m-rank* of a module M over a poset P is a restriction of the multirank

$$m\text{-rank} = \text{multirank}|_{P^m \times P^m} : P^m \times P^m \rightarrow \mathbb{N}$$

I.e., for subsets $S, T \subseteq P$ both of size at most m ,

$$m\text{-rank}_M(S, T) = \text{multirank}_M(S, T) = \text{rank} \left(\bigoplus_{s \in S} M_s \rightarrow \bigoplus_{t \in T} M_t \right)$$

Remark 3.4.2. Multirank is an extension of rank in the sense that when restricting to singleton sets S and T , $\text{multirank} : P \times P \rightarrow \mathbb{N}$ gives the same information as the rank function. Said another way, the rank function (precomposed with the map sending a poset coordinate $\mathbf{p}$ to its corresponding vector space $M_{\mathbf{p}}$) is exactly the 1-rank function.

The m -rank is sufficient to compute the auxiliary geometric invariants when the structure of a poset is simple enough:

Corollary 3.4.3 (Corollary to theorem 3.3.6). *If every point in the parameter space P has indegree and outdegree (definition 2.3.3) bounded above by m , then for a tame P -module m -rank determines the multiplicities of births and deaths at every parameter value.*

Proof. Apply lemma 3.2.8 to the source and target sets of theorem 3.3.6 to create expressions for the numbers of births and deaths in terms of m -rank. The source and target sets are of size $|P_{\preceq \mathbf{p}}|$ and $|P_{\succeq \mathbf{p}}|$, but $P_{\preceq \mathbf{p}}$ can be replaced by the points covered by $\mathbf{p}$ and $P_{\succeq \mathbf{p}}$ can be replaced by the points covering $\mathbf{p}$ (definition 2.3.3). The bound on indegree and outdegree guarantees that these new source and target sets are of size at most m , so they fall in the domain of the m -rank function. $\square$

A similar statement can be made relating m -rank to merge and branch numbers, but since merge (resp. branch) numbers can have source (target) set “far away” from the

target (source), m cannot be chosen just based on indegree (outdegree). A branch (merge) number can be computed using m -rank as long as the source (target) set is of size at most m .

Remark 3.4.4. Computing the value of $\text{multirank}(S, T)$ is computing the rank of a block matrix with blocks $M_s \rightarrow M_t$ for all pairs $(s, t) \in S \times T$. This matrix is size

$$\left(\sum_{s \in S} \dim_{\mathbb{k}} M_s \right) \times \left(\sum_{t \in T} \dim_{\mathbb{k}} M_t \right)$$

For an $m \times n$ matrix with $m \geq n$, Gaussian elimination can be used to compute matrix rank in $O(mn^{\omega-1})$ field operations where $\omega < 2.373$ is the matrix multiplication exponent [BH74; IMH82; CW90; Wil12]. There are also non-deterministic algorithms that are faster in certain circumstances [Wie86; GG03; SSV04].

The computation of an m -rank takes m^{ω} times as many field operation as a 1-rank for square matrices. For example, a 2-rank takes ≈ 5.18 times as many field operations as the computation of a rank.

The number of ranks and m -ranks to compute depends on the size and structure of the poset P . The domain of rank is P^2 and the domain of m -rank is P^{2m} , so a rough estimate is that if there are R -many rank computations, there are R^m -many m -rank computations.

Altogether given that computing the rank functions takes time R , computing the m -rank function might take time $m^{2.373} R^m$. Changes to the poset for speeding up m -rank computations could include reducing the size of the poset to reduce R but also finding a “simpler” poset – with lower maximum in- and out-degrees, perhaps – to reduce the m needed to gather sufficient information.

3.4.2 Example computations

We can use multirank and merge and branch numbers to study a persistence module. Specifically, we can use these three functions to determine whether there are features that meet requirements about their births and deaths.

Example 3.4.5. Suppose $\mathbf{p}, \mathbf{q} \preceq \mathbf{r}$. Note that neither $\mathbf{p}$ nor $\mathbf{q}$ needs to be covered by $\mathbf{r}$.

- 1) The homology classes alive at $M_{\mathbf{p}}$ and also alive at $M_{\mathbf{r}}$ are the ones spanning $\text{im}(M_{\mathbf{p}} \rightarrow M_{\mathbf{r}})$. The dimension of the vector space spanned by these homology classes is $\text{rank}(M_{\mathbf{p}} \rightarrow M_{\mathbf{r}}) = \text{multirank}(\{\mathbf{p}\}, \{\mathbf{r}\})$.
- 2) The homology classes alive at both $M_{\mathbf{p}}$ and $M_{\mathbf{q}}$ and also alive at $M_{\mathbf{r}}$ are the ones spanning

$$\text{im}(M_{\mathbf{p}} \rightarrow M_{\mathbf{r}}) \cap \text{im}(M_{\mathbf{q}} \rightarrow M_{\mathbf{r}})$$

The span of these homology classes has dimension $\text{merge}(\{\mathbf{p}, \mathbf{q}\}, \mathbf{r})$.

- 3) The homology classes alive at both $M_{\mathbf{p}}$ and $M_{\mathbf{q}}$ that die at $M_{\mathbf{r}}$ are the homology classes in the span of

$$\ker \left(M_{\mathbf{r}}|_{\text{im}(M_{\mathbf{p}} \rightarrow M_{\mathbf{r}}) \cap \text{im}(M_{\mathbf{q}} \rightarrow M_{\mathbf{r}})} \rightarrow \bigoplus_{\mathbf{t} \succ \mathbf{r}} M_{\mathbf{t}} \right)$$

which is a vector space of dimension

$$\dim_{\mathbb{k}} M_{\mathbf{r}} - \text{merge}(\{\mathbf{p}, \mathbf{q}\}, P_{\succ \mathbf{r}})$$

Note that for this merge computation we replace $M_{\mathbf{t}}$ by $\bigoplus_{\mathbf{t} \succ \mathbf{r}} M_{\mathbf{r}}$.

Chapter 4

Metrics

Metrics are important tools for analyzing persistence output. They quantify the similarity between modules, and pairwise distances between data are much easier to integrate with other measurement tools than more information-rich but complicated module invariants.

Many persistence “metrics” or “distances” are

- 1) *extended* (meaning they can take the value infinity)
- 2) *pseudometrics* (meaning there are pairs of non-isomorphic modules that are distance 0 from one another).

These two issues come up when constructing L_p spaces (which are spaces of functions) and are dealt with by

- 1) only considering integrable functions (functions with finite norm)
- 2) defining an equivalence relation such that functions of distance 0 from one another are equivalent.

We could choose to make these same restrictions and equivalences for persistence modules but for now we acknowledge the limitations of persistence metrics rather than define the problems away.

Terminology. Throughout this text, when referring to persistence “metrics” and “distances,” we really mean extended pseudometrics. Similarly, any “norms” on per-

sistence modules are assumed to be extended seminorms (definition 4.3.20).

4.1 Known metrics

As mentioned before, most persistence metrics are really *extended pseudometric*.

Definition 4.1.1. An *extended pseudometric* is a function $d : X \times X \rightarrow [0, \infty]$ that satisfies

$$d(x, x) = 0,$$

is symmetric

$$d(x, y) = d(y, x),$$

and satisfies the triangle inequality

$$d(x, z) \leq d(x, y) + d(y, z).$$

It is generally desirable for a metric to be stable so that introducing error or approximations doesn't seriously affect the outcomes of analysis. Many stability theorems are stated in terms of the *sublevelset persistence* construction because that construction encapsulates many common persistence settings, such as persistence from a filtration of a finite simplicial complex [CEH07; CC+09; CD+13; Les15; BSS15].

Construction 4.1.2 (Sublevelset persistence). *Sublevelset persistent homology* is computed from a *filtration function* $\gamma : X \rightarrow P$ for some poset P that acts as the parameter space of the filtration and persistence module. To construct the filtration $\{X_{\mathbf{p}}\}_{\mathbf{p} \in P}$, let $X_{\mathbf{p}} = \gamma^{-1}(P_{\preceq \mathbf{p}}) \subseteq X$ be the preimage in X of all parameters $\mathbf{q} \preceq \mathbf{p} \in P$.

Then the persistent homology is

$$H_i X_\bullet = \bigoplus_{\mathbf{p} \in P} H_i(X_{\mathbf{p}})$$

with structure morphisms induced by inclusions.

Remark 4.1.3. Note that though we omit the function γ from the notation for simplicity, $H_i X_\bullet$ does depend on γ .

Definition 4.1.4. An extended pseudometric (Definition 4.1.1) d on persistence $\mathbb{R}^n$ -modules is *stable* if for any functions $\gamma_X : X \rightarrow \mathbb{R}^n$ and $\gamma_Y : Y \rightarrow \mathbb{R}^n$ on any topological spaces X and Y , and given any integer $i \geq 0$,

$$d(H_i X_\bullet, H_i Y_\bullet) \leq d_\infty(\gamma_X, \gamma_Y).$$

4.1.1 Interleaving distance

The interleaving distance was originally defined for single-persistence [CC+09], then for multipersistence in a 2011 preprint of [Les15], and finally “generalized” persistence – which is persistence over arbitrary preordered sets – where the definition was made functorial [BSS15]. Here we present the definition from [Les15], which focuses on interleavings for $\mathbb{R}^n$ -modules.

We use the notation that if (M, φ) is a P -module then $\varphi_{\mathbf{p} \rightarrow \mathbf{q}} = \varphi_{\mathbf{p}\mathbf{q}} : M_{\mathbf{p}} \rightarrow M_{\mathbf{q}}$ is a structure morphism (definition 2.3.5) for M .

Notation. Let $\epsilon \in \mathbb{R}$. Then the bold character $\boldsymbol{\epsilon}$ denotes the vector defined by

$$\boldsymbol{\epsilon} = (\epsilon, \dots, \epsilon) \in \mathbb{R}^n.$$

Definition 4.1.5. Consider an $\mathbb{R}^n$ -module M . The *shift* of M by $\epsilon \in \mathbb{R}_{\geq 0}^n$ is the module $M(\epsilon)$ such that for all $\mathbf{p} \in \mathbb{R}^n$,

$$M(\epsilon)_{\mathbf{p}} = M_{\mathbf{p}+\epsilon}.$$

Similarly the *shift* of a morphism f is

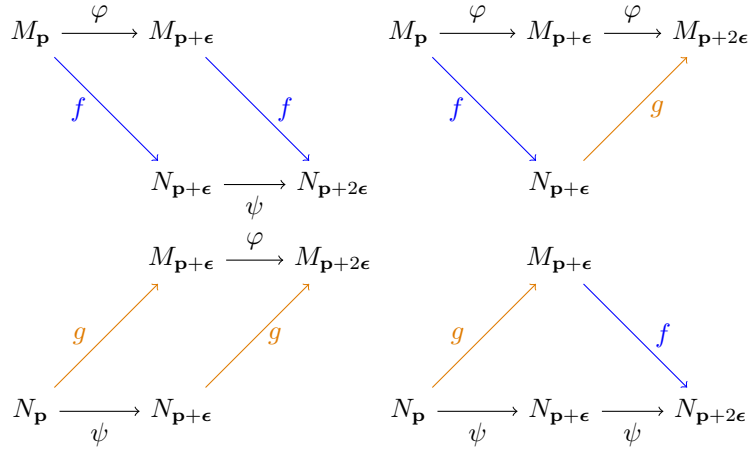
$$f(\epsilon)_{\mathbf{p} \rightarrow \mathbf{q}} = f_{\mathbf{p}+\epsilon \rightarrow \mathbf{q}+\epsilon}.$$

Definition 4.1.6. Modules (M, φ) and (N, ψ) are ϵ -*interleaved* if there exist morphisms $f : M \rightarrow N(\epsilon)$ and $g : N \rightarrow M(\epsilon)$ such that

$$g(\epsilon) \circ f = \varphi_{\bullet \rightarrow \bullet+2\epsilon}$$

$$f(\epsilon) \circ g = \psi_{\bullet \rightarrow \bullet+2\epsilon}.$$

Then f and g are *interleaving morphisms*.



The commutativity of the parallelogram-shaped diagrams above left guarantees f

and g are P -module homomorphisms. The commutativity of the triangular diagrams guarantee that f and g are an ϵ -interleaving.

Definition 4.1.7. The *interleaving distance* is the extended pseudometric on $\mathbb{R}^n$ -modules

$$d_I(M, N) = \inf\{\epsilon \geq 0 \mid M \text{ and } N \text{ are } \epsilon\text{-interleaved}\}.$$

Remark 4.1.8. The existence of a 0-interleaving between two modules implies they are isomorphic but two modules being interleaving distance 0 from one another does not imply they are isomorphic [BS14]. For example a homology class that is born and dies at the same point – a nonzero bar of length 0 in a barcode – is interleaving distance 0 from the 0 module. This length 0 bar is an example of an *ephemeral module*.

Theorem 4.1.9 ([Les15, Corollary 5.6]). *Suppose that for an $\mathbb{R}^n$ -module M , each vector space $M_{\mathbf{p}}$ has scalar field $\mathbb{k}$, a prime field (i.e., $\mathbb{Q}$ or $\mathbb{Z}/p\mathbb{Z}$ for prime p). Then the interleaving distance on $\mathbb{R}^n$ -modules is stable and d_I is universal, i.e. for any other stable metric d on $\mathbb{R}^n$ -modules*

$$d \leq d_I.$$

The requirement on the field $\mathbb{k}$ can be dropped if we restrict to $\mathbb{R}^n$ -modules $H_i X_{\bullet}$ (for a fixed i) as constructed in construction 4.1.2.

Remark 4.1.10. For single-persistence, the interleaving distance is equal to the *bottleneck distance* (originally shown in a preprint of [Les15]).

The bottleneck distance can be described as a distance on persistence diagrams. Each diagram is a collection of points in the plane above the diagonal, with each point representing a homology. The point is plotted according to its homology class's birth and death parameters.

Definition 4.1.11. Let A and B be sets of points in the plane, allowing multiple entries of the same point. Define a *matching* between them by 1) add points on the diagonal $x = y$ to each set under the condition that in the end, A and B have the same cardinality; and 2) defining a bijection $\gamma : A \rightarrow B$.

Then the *bottleneck distance* between single-persistence modules M and N can be defined on their persistence diagrams $\text{Dgm}(M)$ and $\text{Dgm}(N)$:

$$d_B(M, N) = \inf_{\gamma: \text{Dgm}(M) \rightarrow \text{Dgm}(N)} \sup_{a \in \text{Dgm}(M)} \|a - \gamma(a)\|_\infty$$

Remark 4.1.12 (Computing interleaving distance). As of the time of writing the author does not know of a way to efficiently approximate the interleaving distance for multipersistence modules. Exactly computing the interleaving distance between multipersistence modules is NP-hard [BBK18].

For single-persistence the interleaving distance is equal to the bottleneck distance, which can be computed [CEH07].

4.1.2 Matching distance

The *matching distance* uses linear subspaces of the parameter space to compare $\mathbb{R}_{\geq 0}^n$ -modules [BC+08; CDF10]. The modules are *restricted* (definition 4.1.15) to one linear subspace at a time and then compared as single-persistence modules using the bottleneck distance (definition 4.1.11). The definition includes a normalization term that guarantees stability with respect to the L_∞ metric on the filtration function used to construct the sublevelset persistence module (construction 4.1.2). And, just as for the interleaving distance, the matching distance on single-persistence is equivalent to the

bottleneck distance.

Terminology. Many references on matching distance use different terminology than is used in this dissertation. For example, the rank function is sometimes referred to as *persistent Betti numbers* and the bottleneck distance is referred to as the *single dimensional matching distance*. The matching distance is also defined for *size functions*, which are rank functions of 0th homology persistence modules and are the topic of study in [BC+08].

The first step in defining the multipersistence matching distance is reparameterizing $\mathbb{R}_{\geq 0}^n$ -modules so that they can be studied along linear subspaces of their parameter space. Instead of considering parameters along the axes of $\mathbb{R}^n$, study the module along rays in the direction of $\vec{l} \in \mathbb{R}^n$ that start at offset $\vec{b} \in \mathbb{R}^n$.

Definition 4.1.13. A pair $(\vec{l}, \vec{b}) \in \mathbb{R}^n \times \mathbb{R}^n$ is *admissible* if $\vec{l} = (l_1, \dots, l_n)$ is a unit vector with all positive entries and $\vec{b} = (b_1, \dots, b_n)$ lies on the hyperplane through the origin with normal vector $(1, \dots, 1)$, which is exactly the hyperplane $\{\vec{x} \mid \vec{x} \cdot (1, \dots, 1) = 0\}$. The requirement that $\vec{b}$ lie on this plane is equivalent to requiring $\sum_{i=1}^n b_i = 0$.

Remark 4.1.14. The vector $\vec{l}$ must be a unit vector, but that can be with respect to any L_p norm with integer $p \geq 1$. That means there are countably many different ways to choose the set of admissible pairs. All of the matching distances that come from these different admissible sets are the same, but computations seem to be easiest when $p = 1$ [CF10].

For each admissible pair $(\vec{l}, \vec{b})$ there is a ray $\{\lambda \vec{l} + \vec{b} \mid \lambda \in \mathbb{R}_{\geq 0}\}$. It is along these rays that we restrict our multipersistence modules and compute the bottleneck distance. The closure of the collection of rays $\{\lambda \vec{l} + \vec{b} \mid \lambda \in \mathbb{R}_{\geq 0}\}_{(\vec{l}, \vec{b})}$ covers the support of

any $\mathbb{R}_{\geq 0}$ -module M , and also covers every $\mathbf{p} \preceq \mathbf{q} \in \mathbb{R}^n$ in the domain of the rank function of M with a ray through both $\mathbf{p}$ and $\mathbf{q}$. This implies that a reparameterization of an $\mathbb{R}_{\geq 0}^n$ -module into single-persistence modules indexed by admissible pairs $(\vec{l}, \vec{b})$ encapsulates at least most of the information of the rank function.

Definition 4.1.15. Fix an $\mathbb{R}_{\geq 0}^n$ -module M . Define $M_{(\vec{l}, \vec{b})}$ to be an $\mathbb{R}_{\geq 0}$ -module with parameter $\lambda \in \mathbb{R}_{\geq 0}$ via

$$\left(M_{(\vec{l}, \vec{b})}\right)_\lambda = \begin{cases} M_{\lambda\vec{l} + \vec{b}} & \lambda\vec{l} + \vec{b} \in \mathbb{R}_{\geq 0}^n \\ 0 & \lambda\vec{l} + \vec{b} \notin \mathbb{R}_{\geq 0}^n. \end{cases}$$

We call $M_{(\vec{l}, \vec{b})}$ the *restriction* of M to the ray $\{\lambda\vec{l} + \vec{b} \mid \lambda \in \mathbb{R}_{\geq 0}\}$.

Definition 4.1.16. Let M and N be $\mathbb{R}_{\geq 0}^n$ -modules. Then the *matching distance* between M and N is

$$d_{\text{match}}(M, N) = \sup_{(\vec{l}, \vec{b}) \in \text{Adm}_n} \min_{i=1, \dots, n} l_i d_B \left(M_{(\vec{l}, \vec{b})}, N_{(\vec{l}, \vec{b})} \right)$$

where d_B is the bottleneck distance.

The normalization (multiplication by the smallest l_i) is sufficient to guarantee stability [BC+08; CDF10].

Remark 4.1.17 (Computing matching distance). The matching distance can be restricted to taking a maximum over a finite number of admissible pairs instead of a supremum over all admissible pairs. There is an algorithm for approximating the matching distance for 2-persistence within a given error bound by finding an appropriate finite set of admissible pairs [BC+11].

Remark 4.1.18 (An extension of the matching distance). There is a *coherent match-*

ing distance for 2-persistence that is also stable and contains the same information as the original matching distance. The matchings from the bottleneck distance on restrictions of each module change continuously with respect to the filtration. One difficulty of defining coherency was that multipersistence can have monodromy, which is studied in [CEF13].

Since the matching distance is computed from a restriction of the rank function it does not contain any more information than the rank function itself. It does, however, have stability with respect to L_∞ perturbations of the filtering function.

4.1.3 Existing L_p distances

There are a few distances based on using L_p distances for invariants that are functions from the parameter space $\mathbb{R}^n$ (or copies of the parameter space) to $\mathbb{R}$. Examples include

- 1) the Hilbert function of a module (definition 3.1.3), which maps a point in the parameter space to the dimension of the vector space at that point. The L_p distance on the Hilbert function of $\mathbb{R}^n$ -modules is used for multipersistence applications like [KLW18] and has been discussed in [LW15; HO+17].
- 2) the rank function (definition 3.1.4), which was used with the L_2 metric to implement PCA on sets of single-persistence modules in [RT16].
- 3) persistence landscapes, which are reparameterizations of the rank function that guarantee a stability result [Bub15; Vip18].
- 4) persistence images, which use a weighting function to deliver a stability result [AE+17].

In [RT16] Robins and Turner consider single-persistence modules that have classes born at $-\infty$ or dying at $+\infty$. They note that the rank distance is not finite if the two modules have different numbers of births at $-\infty$ or deaths at $+\infty$.

This brings up the question: How should we compare functional module invariants whose supports don't have finite measure?

- 1) The measure can be changed by a weighting function as in [RT16], but this can't account for all instances. The reason is based on a requirement for the weighting function: it should have translation invariance so that homology classes with the same persistence but different birth times contribute the same amount to the distance. This translation invariance allows a homology class that has existed and will exist for all parameter values to give an infinite contribution to a difference of rank functions.
- 2) These modules with unbounded support can be truncated, i.e., all vector spaces and maps outside a bounded set in the parameter space are set to 0. The software RIVET [KLW14; LW15] does this in a non-arbitrary way by truncating a module outside the smallest rectangle containing the origin and all nonzero Betti numbers. This part of the module determines the entire module if the module is finitely presented, but one has to be able to guarantee that all nonzero Betti numbers have in fact been found.

Joint work with Ezra Miller in section 4.3 shows a way to modify existing distances to account for immortal homology classes in a canonical way. The idea is to separate out the parts of the module that are immortal, compare the mortal and immortal parts separately, and avoid integrating over unbounded domains. The immortal parts must themselves be sorted so that only homology classes with the same *pattern of*

immortality – the lists of directions along which a homology class is immortal or mortal – are compared with one another. This technique can even compare modules with different numbers of immortal homology classes.

4.2 New L_p metrics

The distances in subsection 4.1.3 can all be defined as L_p -metrics for arbitrary p , though some stability or statistical results may not be preserved when p changes. The next subsection describes some qualitative features of L_p distances on multipersistence and the following subsection describes an L_p metric on m -rank.

4.2.1 Qualitative features of L_p distances

The L_p distances behave differently than the interleaving and matching distances. To show why, we first explicitly define the L_p rank distance:

Definition 4.2.1. The *rank poset*, which will be discussed more later in subsection 4.3.5, is the set

$$P \times_{\preceq} P = \{(\mathbf{p}, \mathbf{q}) \in P \times P \mid \mathbf{p} \preceq \mathbf{q}\}$$

with partial order inherited from $P \times P$, i.e.

$$(\mathbf{p}, \mathbf{q}) \preceq (\mathbf{r}, \mathbf{s}) \iff \mathbf{p} \preceq \mathbf{r} \text{ and } \mathbf{q} \preceq \mathbf{s}.$$

Definition 4.2.2. The L_p rank distance between $\mathbb{R}^n$ -modules M and N is

$$d_{\text{rank}}(M, N) = \|\text{rank}_M - \text{rank}_N\|_p = \left(\int_{\mathbb{R}^n \times_{\preceq} \mathbb{R}^n} |\text{rank}_M - \text{rank}_N|^p \right)^{1/p}$$

where $\|\cdot\|_p$ is the usual L_p norm on $\mathbb{R}^{2n}$ restricted to $\mathbb{R}^n \times_{\preceq} \mathbb{R}^n$.

In single-persistence, where the interleaving and matching distances are equivalent to the bottleneck distance, the features that the L_p rank distance ($p > 1$) and bottleneck distance consider noise are different. For the bottleneck distance any sufficiently short bar in a barcode is noise, no matter its relation to other bars. This corresponds to a point close to the diagonal on a persistence diagram.

For the L_p rank distance ($p > 1$), however, if many short bars are overlapping over a set of positive measure in the parameter space, then the exponent p in the definition of rank distance amplifies the contribution of these bars. The rank distance doesn't consider all sufficiently short bars as noise; it matters how bars' lifetimes overlap.

Features of both the bottleneck and rank distances are illustrated in figure 4.2. There are two barcodes for single-persistence modules, each having the same number of “short” bars. Each of these bars, on its own, is small enough to be considered noise by both the bottleneck and rank distances. As p increases, the rank distance assigns more signal to the *stack* of bars.

barcode		
bottleneck distance	noise	noise
rank distance ($p = 1$)	noise	noise
rank distance ($p > 1$)	noise	signal

Figure 4.2. Differences between bottleneck and L_p rank distances on single-persistence module barcodes. As p increases the rank distance assigns more signal to clusters of short bars.

The behavior of detecting when many features exist at the same scale may be useful for applications looking to detect consistency, repetitiveness, or regularity. For example, normal cells are very regular while cancer cells are not, so the ability to detect stacking behavior might be able to help automation of detecting or grading cancers, like in [LB+17].

4.2.2 L_p m -rank metric

Since multirank and m -rank are functional module invariants whose domains are sums of copies of the parameter space, they, too, give L_p metrics. Since the multirank metric has a prohibitively large domain we just give the definition of the m -rank metric.

Recall the m -rank from definition 3.4.1: Let M be an $\mathbb{R}^n$ -module. For subsets $S, T \subseteq \mathbb{R}^n$ of size at most m , the m -rank : $(\mathbb{R}^n)^{2m} \rightarrow \mathbb{N}$ is

$$m\text{-rank} : (S, T) \mapsto \text{rank} \left(\bigoplus_{s \in S} M_s \rightarrow \bigoplus_{t \in T} M_t \right)$$

For m -rank to fit into an L_p metric the sets S and T must be ordered and of size m , so add null values ($\emptyset$) to the sets until they are the appropriate size. The convention is that any map from or to a vector space indexed by a null value is 0.

Definition 4.2.3. The L_p m -rank distance between $\mathbb{R}^n$ -modules M and N is

$$d_{m\text{-rank}}(M, N) = \|m\text{-rank}_M - m\text{-rank}_N\|_p = \left(\int_{\mathbb{R}^{2mn}} |m\text{-rank}_M - m\text{-rank}_N|^p \right)^{1/p}$$

where $\|\cdot\|_p$ is the usual L_p distance on $\mathbb{R}^{2mn}$.

4.3 Primary decomposition distances

The work in this section was done with Ezra Miller and will be available in [MT19].

The point of primary decomposition in persistence is to separate out parts of a persistence module that are immortal in different directions. For example in single-persistence, when the parameter takes values in $\mathbb{R}_{\geq 0}$ then homology classes can either have finite lives or be immortal. Similarly, in two-persistence with nonnegative parameters a class can be mortal, immortal in the x direction, immortal in the y direction, or immortal in both directions.

These *mortality patterns* are what makes measuring distance between modules difficult: Modules that have classes with different mortality patterns are often infinite distance apart. In [RT16], Robins and Turner note that to get a finite L_p rank distance between two single-persistence modules the modules must have the same number of classes living to $-\infty$ as well as the same number living to $+\infty$.

To fix this problem, we propose decomposing modules into components with different

mortality patterns and defining a distance on each of those components. A weighted sum of component-wise distances gives a distance between two modules.

Notice that primary decomposition is a technique to modify *existing* distances. Below, we develop the decomposition modification for L_p distances on the Hilbert functions of modules, but other distances can be substituted. The L_p Hilbert distances and the L_2 Hilbert norm are discussed in subsection 5.4.3 as part of future analysis for the fly wing application in chapter 5.

This strategy does not define finite distances between all pairs of modules – see example 4.3.28 – so the idea of a module being *facially bounded* (definition 4.3.29) is developed as sufficient condition for being of finite distance from the 0 module. The idea of a *length* for an $\mathbb{R}^n$ -module is also developed.

4.3.1 Primary decomposition

This section restates classical primary composition in terms of modules over Q , a *polyhedral partially ordered group* (definition 4.3.1). Q , which is frequently $\mathbb{Z}^n$ or $\mathbb{R}^n$, takes over the role of parameter space poset here but its geometry gives it much more structure than an arbitrary poset. Details on primary decomposition for polyhedral partially ordered groups can be found in [Mil17, Chapter 3]. A reference for classical primary decomposition is [Eis95, Chapter 3].

We use the idea of polyhedral faces to specify the “directions” (or axes) along which an element in a module lives to infinity.

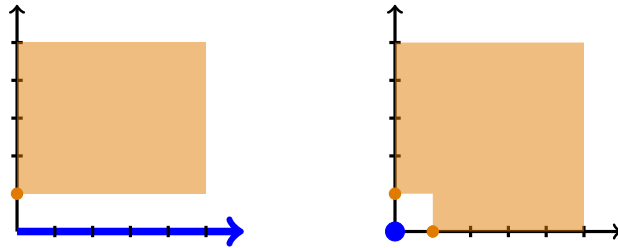
Definition 4.3.1. Let Q be an abelian group.

- 1) Q_+ is a *positive cone* of Q if Q_+ is a submonoid with trivial unit group.
- 2) Q is *partially ordered* if it is generated (as a group) by its positive cone Q_+ .
- 3) An *ideal* is a subset $I \subseteq Q$ such that $I + Q = I$.
- 4) A *face* of positive cone Q_+ is a submonoid $\tau \subseteq Q_+$ such that $Q_+ \setminus \tau$ is an ideal of the monoid Q_+ .
- 5) Q is *polyhedral* (i.e., Q is a *polyhedral partially ordered group*) if it has only finitely many faces.

Example 4.3.2 ([MS05, Definition 7.8]). $\mathbb{Z}^n$ is a polyhedral partially ordered group with positive cone $\mathbb{N}^n$ and $\mathbb{Z}^n$ -modules can be thought of as modules over a polynomial ring $\mathbb{k}[x_1, \dots, x_n]$. A subset $I \subseteq \mathbb{N}^n$ is an ideal if $I + \mathbb{N}^n \subseteq I$. A submonoid τ of $\mathbb{N}^n$ is a face if the complement $\mathbb{N}^n \setminus \tau$ is an ideal. In this setting, a subset $\tau \in \mathbb{N}^n$ is a face if and only if it is a coordinate subspace of $\mathbb{N}^n$. The geometric intuition that the faces of the first quadrant are the faces of $\mathbb{N}^n$ as a positive cone in a polyhedral partially ordered group carries over into the case of polyhedral partially ordered group $\mathbb{R}^n$ with positive cone $\mathbb{R}_{\geq 0}^n$.

Below left is the ideal $\langle y \rangle$ of the polyhedral partially ordered group $\mathbb{k}[x, y]$. The face associated to $\langle y \rangle$ is the x -axis, which can also be written as $(1, 0)$.

Below right is the maximal ideal $\langle x, y \rangle$ with associated face $(0, 0)$ (the 0-face).



Terminology. We may refer to a face as $\tau \subseteq \{1, \dots, n\}$ to record the “directions” that specify a face F . The complement of a face τ is another face $\bar{\tau} = \{1, \dots, n\} \setminus \tau$.

Lemma 4.3.3 ([Mil17, Lemma 3.4]). *A module over a polyhedral partially ordered group Q with positive cone Q_+ is a Q -graded module over the monoid algebra $\mathbb{k}[Q_+]$.*

Definition 4.3.4. Let τ be a face of the positive cone Q_+ of the polyhedral partially ordered group Q . An element in a Q -module is

- τ -persistent if it lives when pushed up along any combination of τ -axes;
- $\bar{\tau}$ -transient if it dies when pushed up sufficiently along any non- τ axis;
- τ -coprimary if it is both τ -persistent and $\bar{\tau}$ -transient.

A Q -module is τ -coprimary if every element divides a τ -coprimary element.

Theorem 4.3.5 (Primary decomposition [Mil17, Definition 3.28]). *Finitely encoded $\mathbb{R}^n$ -modules admit primary decompositions:*

$$M \hookrightarrow \bigoplus_{i=1}^r M/M_i$$

where the M/M_i are τ -coprimary quotients of M .

Definition 4.3.6. The τ -coprimary quotients M/M_i in the primary decomposition of M are called τ -coprimary components of M .

Remark 4.3.7. Primary decomposition is usually stated in terms of primary submodules $M_i \subseteq M$ such that

$$\bigcap_{i=1}^r M_i = 0 \text{ in } M$$

The primary submodules M_i are defined as having coprimary quotients M/M_i .

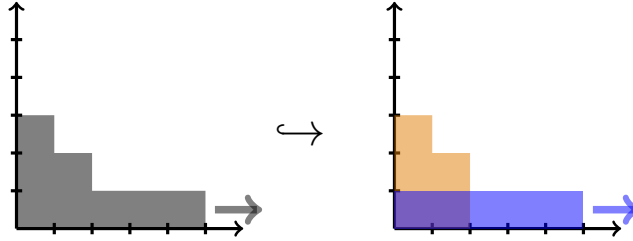
Proposition 4.3.8. *The definitions of primary decomposition given in remark 4.3.7*

and theorem 4.3.5 are equivalent.

Proof. Consider the morphism from theorem 4.3.5. The kernel of this morphism is the intersection of the M_i . The morphism is injective if and only if the intersection is 0 in M . $\square$

Example 4.3.9.

- 1) The 0-coprimary elements of a module are the elements that die when pushed far enough in any direction. A finite length $\mathbb{Z}^n$ module is 0-coprimary.
- 2) Below left is a module where the arrow indicates that the elements persist along the x -axis immortally. On the right is (the direct sum of) a 0-coprimary component and a $(1,0)$ -coprimary component. The injection is a primary decomposition, as in theorem 4.3.5.



Since we want to use primary decomposition for a distance, it would be nice to find a canonical primary decomposition to use so the distances are well-defined; however, there is no canonical primary decomposition for $\mathbb{R}^n$ -modules. On the other hand there is a sense of minimality.

Recall that a socle (definition 3.3.2) is essentially a collection of deaths. The *socle along a face* soc_τ is a collection of deaths with a mortality pattern determined by τ : immortal in the τ directions and mortal in the $\bar{\tau}$ directions.

Definition 4.3.10 ([Mil17, Section3.3]). A primary decomposition $M \hookrightarrow \bigoplus_{i=1}^r M/M_i$ is *minimal* if for $i = 1, \dots, r$,

$$\mathrm{soc}_\tau M \xrightarrow{\sim} \mathrm{soc}_\tau(M/M_i).$$

In a minimal primary decomposition there is exactly one component M/M_i for each of M 's associated primes (see [Eis95, Chapter 3] for details).

4.3.2 Extracting components of a module

How do we compute the coprimary components in example 4.3.9 (2)? There are a few steps needed to extract a τ -coprimary component of a module M . Namely, the elements that are not τ -persistent and those that are not $\bar{\tau}$ -transient must be ignored.

First we take the localization of M , which will invert the elements in the direction of τ .

Definition 4.3.11 ([Mil17, Definition3.23]). Fix a face τ of a polyhedral partially ordered group Q . The *localization* of a Q -module M along τ is the tensor product

$$M_\tau = M \otimes_{\mathbb{k}[Q_+]} \mathbb{k}[Q_+ + \mathbb{Z}\tau]$$

In the natural homomorphism from M to the localization M_τ , any element of M that dies when pushed along τ maps to 0, as in example 4.3.19. So $M \rightarrow M_\tau$ maps all elements that are not τ -persistent to 0.

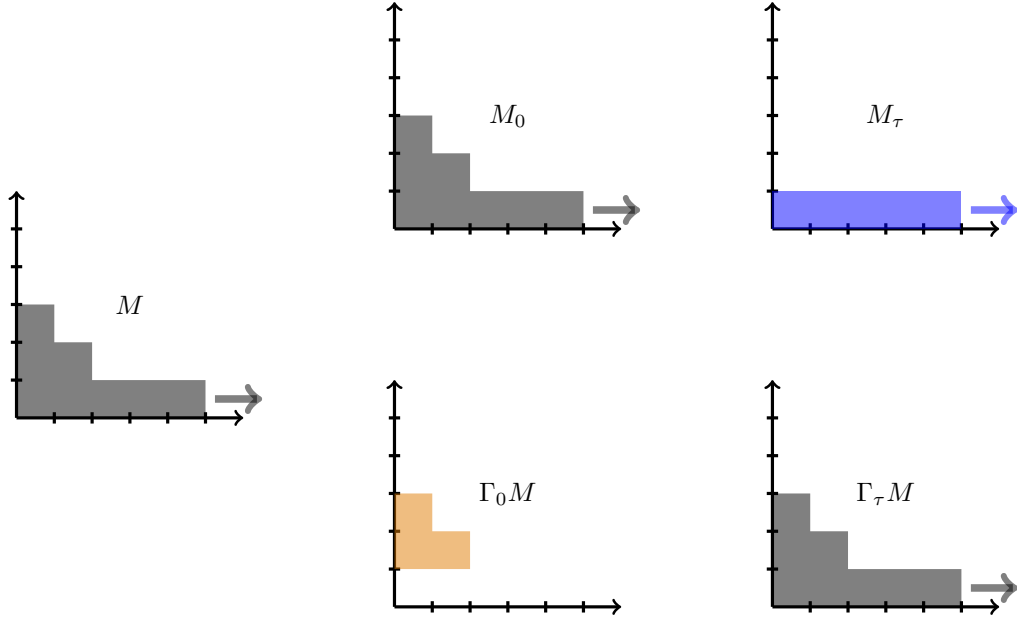
Definition 4.3.12 ([Mil17, Definition3.23]). The submodule of M *globally supported*

on τ is

$$\Gamma_{\tau}M = \ker \left(M \rightarrow \bigoplus_{\tau' \not\leq \tau} M_{\tau'} \right)$$

$\Gamma_{\tau}M$ ignores the elements of M that are not $\bar{\tau}$ -transient.

Example 4.3.13. Revisit the module M from Example 4.3.9(2). Below are localizations and global supports for the faces 0 and $\tau = \text{the } x\text{-axis}$.



Proposition 4.3.14 ([Mil17, Proposition 3.27]). *For modules over a polyhedral partially ordered group, localization commutes with taking support. That is,*

$$(\Gamma_{\tau'}M)_{\tau} = \Gamma_{\tau'}(M_{\tau})$$

Definition 4.3.15. Let M be a module over a polyhedral partially ordered group Q . The *local τ -support* of M is the module $\Gamma_{\tau}M_{\tau}$ consisting of elements globally supported on τ in the localization M_{τ} .

Definition 4.3.16. Let M be a module over a polyhedral partially ordered group Q .

The *reduced local τ -support* of M is

$$\Gamma_\tau M / \tau = (\Gamma_\tau M_\tau) / \tau$$

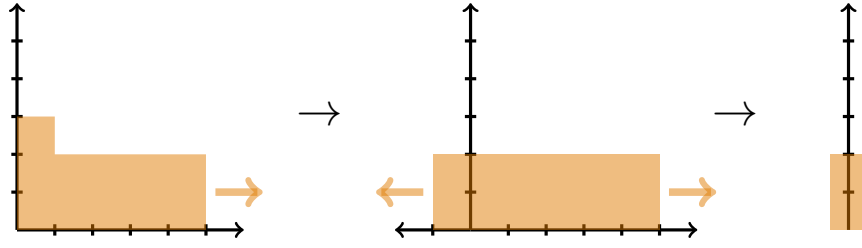
in which $\Gamma_\tau M_\tau$ is collapsed modulo the free action of the group $\mathbb{Z}\tau$ (as in [Mil17, Definition 6.35]).

Remark 4.3.17 (A remark on terminology). The term “reduced local support” will most likely be changed in the near future.

Remark 4.3.18. If M is a τ -coprimary $\mathbb{Z}^n$ -module then taking the reduced local support can take an infinite-length module M to a finite-length module $\Gamma_\tau M / \tau$. It is this operation that makes primary decomposition useful for defining distances: some pairs of modules previously considered infinite distance apart can be primary-decomposed in a way such that a finite distance between them can be defined. This also applies to $\mathbb{R}^n$ -modules, with the appropriate definition of length (see definition 4.3.23).

Example 4.3.19. Let τ be the x -axis-face.

- 1) Start with module M below left.
- 2) Take the localization along the x -axis to get the middle module M_τ .
- 3) Taking the global τ -support of M_τ does nothing: If we localize by the 0 -face or the y -axis-face (the two faces not containing τ), in both instances the y -axis gets inverted and all of M_τ goes to 0 . So $\Gamma_\tau M_\tau = M_\tau$.
- 4) Finally, we mod out by the action of τ to get the reduced local support $\Gamma_\tau M / \tau$ on the right. Notice that while M had infinite length, $\Gamma_\tau M / \tau$ has finite length.



4.3.3 Metrics and norms

The following are definitions and lemmas about extended pseudometrics and their norm-like counterparts. They are used in the following subsection.

Recall that an extended pseudometric is a positive semidefinite, symmetric function $d : X \times X \rightarrow [0, \infty]$ that satisfies the triangle inequality (definition 4.1.1).

Definition 4.3.20. An *extended seminorm* on a vector space X is a function $\|\cdot\| : X \rightarrow [0, \infty]$ that satisfies the triangle inequality,

$$\|x + y\| \leq \|x\| + \|y\|,$$

and is absolutely homogeneous:

$$\|\alpha x\| = |\alpha| \|x\|$$

for $x, y \in X$ and $\alpha \in \mathbb{R}$. By convention the product $0(+\infty)$ is defined to be 0.

Lemma 4.3.21. An extended seminorm $\|\cdot\| : X \rightarrow [0, \infty]$ induces an extended pseudometric $d : X \times X \rightarrow [0, \infty]$ via

$$d(x, y) := \|x - y\|$$

for $x, y \in X$.

Proof. $d(x, y) = \|x - y\|$ is an extended pseudometric: It is immediate to check that absolute homogeneity guarantees symmetry and positive semidefiniteness and that the triangle inequality for the norm guarantees the triangle inequality for the metric.

- Positive semidefinite: $d(x, y) = \|x - y\| \geq 0$; $d(x, x) = \|x - x\| = \|0\| = 0$.
- Symmetric: $d(x, y) = \|x - y\| = |-1|\|y - x\| = d(y, x)$.
- Satisfies triangle inequality:

$$d(x, z) = \|(x - y) - (z - y)\| \leq \|x - y\| + \|y - z\| = d(x, y) + d(y, z).$$

We need to be careful about any of the expressions above being undefined because of subtracting infinity from infinity. But because the norms can only take the value of positive infinity and none of the statements involve subtraction, that will not occur. □

Lemma 4.3.22.

- 1) *A nonnegative linear combination of extended seminorms is an extended seminorm.*
- 2) *A nonnegative linear combination of extended pseudometrics is an extended pseudometric.*

Proof.

- 1) Requiring that the linear combination have positive coefficients guarantees that the nonnegativity of the codomain and the property of absolute homogeneity (specifically, that a scalar's only contribution to a norm is its amplitude) are preserved.

Let $\|\cdot\| = \sum_{i=1}^n \omega_i \|\cdot\|_i$ with $\omega_i \geq 0$.

$$\|x + y\| = \sum_{i=1}^n \omega_i \|x + y\|_i \leq \sum_{i=1}^n \omega_i \|x\|_i + \sum_{i=1}^n \omega_i \|y\|_i = \|x\| + \|y\|$$

Notice that the scalars ω_i must be nonnegative so that the inequality holds.

- 2) It is immediate to check that symmetry is preserved by linear combination of extended pseudometrics. Requiring that linear combinations have nonnegative coefficients guarantees that the codomain, positive semi-definiteness, and triangle inequality are preserved as well.

□

4.3.4 Primary-component-wise norms and distances

In this section M is a Q -module where Q is a polyhedral partially ordered group with positive cone Q_+ . Most of the time $Q = \mathbb{R}^n$ with $Q_+ = \mathbb{R}_{\geq 0}^n$ or $Q = \mathbb{Z}^n$ with $Q_+ = \mathbb{N}^n$.

Finite length modules

The idea of finite length for $\mathbb{Z}^n$ -modules is well understood: the length is the module's dimension as a vector space over $\mathbb{k}$. If this definition is extended to $\mathbb{R}^n$ -modules, however, even those modules with support bounded in $\mathbb{R}^n$ would have infinite length. To instead make sure a pointwise finite-dimensional $\mathbb{R}^n$ -module with bounded support is considered finite length, we introduce a different definition.

Definition 4.3.23. The *p-length* of a 0-coprimary $\mathbb{R}^n$ -module M is the extended

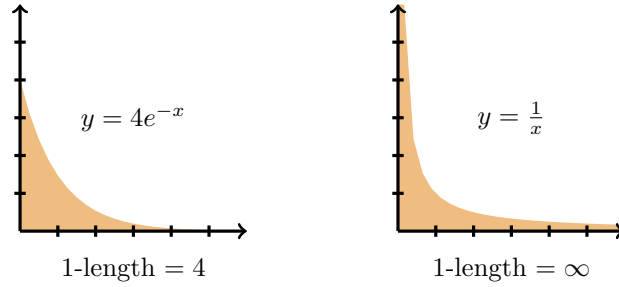
seminorm

$$\|M\|_p = \left(\int_{\mathbb{R}^n} \text{Hilb}_M^p \right)^{1/p} = \|\text{Hilb}_M\|_{L_p}$$

where $\|\cdot\|_{L_p}$ is the usual L_p norm on $\mathbb{R}^n$.

Remark 4.3.24. For $\mathbb{Z}^n$ modules, 0-coprimary pointwise finite-dimensional modules correspond to finite length modules and the 1-length (when integrating with respect to the counting measure) coincides with the usual length of a $\mathbb{Z}^n$ -module.

Example 4.3.25. Below are $\mathbb{R}_{\geq 0}^2$ -modules and their L_1 -lengths. Neither module has bounded support, but the module on the left still has finite 1-length. Both of these modules are 0-coprimary modules.



The L_p -length is an extended seminorm (definition 4.3.20) and gives rise to an extended pseudometric (definition 4.1.1).

Definition 4.3.26. The L_p -Hilbert distance between two $\mathbb{R}^n$ -modules M and N is

$$d_{p\text{-Hilb}}(M, N) = \left(\int_{\mathbb{R}^n} |\text{Hilb}_M - \text{Hilb}_N|^p \right)^{1/p} = \|\text{Hilb}_M - \text{Hilb}_N\|_{L_p}$$

where Hilb is the Hilbert function of a module and $\|\cdot\|_{L_p}$ is the usual L_p norm on $\mathbb{R}^n$.

For facially bounded modules

The extended seminorms from above give rise to extended pseudometrics, which are described here.

Definition 4.3.27. The *facial components* of M are the modules $\Gamma_\tau M/\tau$ for all faces τ in the positive cone Q_+ .

Example 4.3.28. The $\mathbb{R}_{\geq 0}^2$ -modules from example 4.3.25 are both 0-coprimary so their each have only one facial component: themselves. The module to the left is facially bounded and the one to the right is not.

Definition 4.3.29. an $\mathbb{R}^n$ -module M is *facially p -bounded* if all of its facial components are of finite p -length.

Definition 4.3.30. The (weighted) L_p -length of an $\mathbb{R}^n$ -module M is

$$\|M\|_{p\text{-Hilb}} = \sum_{\text{faces } \tau} w_\tau \left(\int_{\mathbb{R}^n} \text{Hilb}_{\Gamma_\tau M/\tau}^p \right)^{1/p} = \sum_{\text{faces } \tau} w_\tau \|\text{Hilb}_{\Gamma_\tau M/\tau}\|_{L_p}$$

where $\|\cdot\|_{L_p}$ is the usual L_p norm on $\mathbb{R}^n$ and the weights w_τ are nonnegative real numbers.

For each face τ consider the reduced local supports $\Gamma_\tau M/\tau$ and $\Gamma_\tau N/\tau$. These are finite p -length (that's the facially bounded criterion) $\mathbb{R}\bar{\tau}$ -modules. Compare their Hilbert functions facial-component-wise using an L_p distance and weights w_τ . This gives a new distance.

Definition 4.3.31. Define the (weighted) L_p -Hilbert distance between two facially

bounded, $\mathbb{R}^n$ -modules M and N .

$$\begin{aligned} d_{p\text{-Hilb}}(M, N) &= \sum_{\text{faces } \tau} w_\tau \left(\int_{\mathbb{R}\bar{\tau}} |\text{Hilb}_{\Gamma_\tau M/\tau} - \text{Hilb}_{\Gamma_\tau N/\tau}|^p \right)^{1/p} \\ &= \sum_{\text{faces } \tau} w_\tau \|\text{Hilb}_{\Gamma_\tau M/\tau} - \text{Hilb}_{\Gamma_\tau N/\tau}\|_{L_p(\mathbb{R}\bar{\tau})} \end{aligned}$$

The L_p norms in the above definitions are norms over $\mathbb{R}\bar{\tau}$, which are Banach spaces of different dimensions. This may create a problem in applications with adding quantities measured with different units, but the weights can be adjusted to account for that discrepancy. The weights ω_τ might be tuned to account for different units, or to guarantee different statistical properties of the distance.

Because of the L_p setting, differences between modules that occur on sets of measure 0 are not detected. That's why we have *pseudometric* and *seminorm*. We have *extended* pseudometric and seminorm because the integrals are not always finite.

Theorem 4.3.32.

- 1) *The L_p -Hilbert lengths $\|\cdot\|_{p\text{-Hilb}}$ defined above are extended seminorms.*
- 2) *The L_p -Hilbert distances $d_{p\text{-Hilb}}$ defined above are extended pseudometrics.*

Proof. Both statements follow from lemma 4.3.22. □

4.3.5 The graph and rank modules

Consider a Q -module M . We construct two auxiliary modules: the graph and rank modules. These modules record information about the maps in M in a new way. They also provide an avenue for using the primary decomposition modification on a

distance based on the ranks of structure morphisms, which is a common measure of a persistence module.

Recall definition 4.2.1: Let Q_+ be the positive cone of a polyhedral partially ordered group Q . Then the *rank poset* of Q is

$$Q \times_{\preceq} Q = \{(\mathbf{p}, \mathbf{q}) \in Q \times Q \mid \mathbf{p} \preceq \mathbf{q}\}.$$

Start with a persistence module H . Denote the structure morphisms of H by $\varphi_{\mathbf{p}\mathbf{q}} : H_{\mathbf{p}} \rightarrow H_{\mathbf{q}}$. Sometimes we will denote them together: (H, φ) is a Q -module with the structure morphisms explicitly named.

Definition 4.3.33. The *graph module* G of a Q -module (H, φ) is a $(Q \times_{\preceq} Q)$ -module where for $\mathbf{p} \preceq \mathbf{q}$ in Q ,

$$G_{\mathbf{p}\mathbf{q}} = \{(x, \varphi_{\mathbf{p}\mathbf{q}}(x)) \mid x \in H_{\mathbf{p}}\} \subseteq H_{\mathbf{p}} \times H_{\mathbf{q}}$$

is the graph of $\varphi_{\mathbf{p}\mathbf{q}} : H_{\mathbf{p}} \rightarrow H_{\mathbf{q}}$. Each morphism

$$G_{\mathbf{p}\mathbf{q}} \rightarrow G_{\mathbf{r}\mathbf{s}}$$

is induced by $\varphi_{\mathbf{p}\mathbf{r}} \times \varphi_{\mathbf{q}\mathbf{s}} : H_{\mathbf{p}} \times H_{\mathbf{q}} \rightarrow H_{\mathbf{r}} \times H_{\mathbf{s}}$.

Definition 4.3.34. The *rank module* R of a Q -module (H, φ) is a $(Q \times_{\preceq} Q)$ -module constructed by projecting the graph module of H onto its second component. That is, for $\mathbf{p} \preceq \mathbf{q} \in Q$,

$$R_{\mathbf{p}\mathbf{q}} = \text{im } \varphi_{\mathbf{p}\mathbf{q}} \subseteq H_{\mathbf{q}}$$

The morphism $R_{\mathbf{pq}} \rightarrow R_{\mathbf{rs}}$ is the restrictions of $\varphi_{\mathbf{qs}}$ to $\text{im } \varphi_{\mathbf{pq}}$.

Proposition 4.3.35. $\dim_{\mathbb{k}} R_{\mathbf{pq}} = \text{rank } \varphi_{\mathbf{pq}}$.

The rank module can be analyzed by taking its primary decomposition and then using an L_p Hilbert function distance on each component. Further details are research in progress.

Chapter 5

Application: topological evolution in fly wing venation

This project studies the relationship between evolutionary continuous and topological morphological changes via vein morphology in fruit fly wings. A population of flies evolves the continuous morphological changes of interest when it develops, on average, wings with *landmarks* located in more extreme locations than occur in “normal” – or *wild type* – flies. See figure 5.2 for examples of continuous variation in landmark locations across species in the Drosophilinae subfamily (*Drosophila melanogaster* is one of the species represented in the figure and is also the model organism used for the experiments discussed here).

Discrete and topological morphological abnormalities occur in populations of wild type flies under no selective pressure. We only attribute these abnormalities to evolution when the rate of the abnormalities changes. See figure 5.1 for examples of topological abnormalities in fly wing venation.

Biologist collaborators at the Houle Lab at Florida State University hypothesize that selection for particular continuous morphological changes correlates with increased rates of topological morphological abnormality (see subsection 5.2.2 for more detailed hypotheses). The goal of the author and collaborators Surabhi Beriwal and Ezra Miller is to quantify both continuous and topological variation in fly wing venation so their relationship can be effectively studied via statistics.

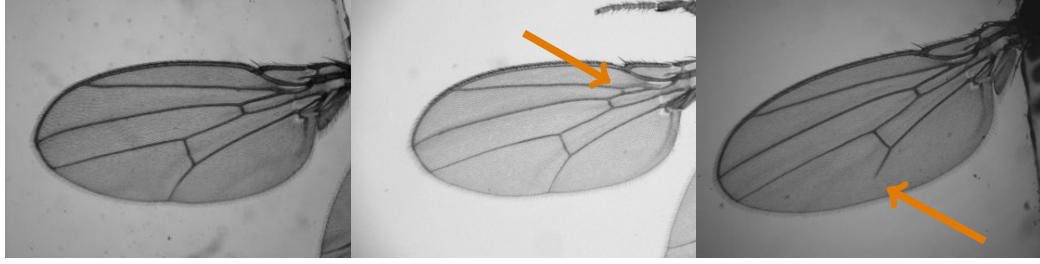


Figure 5.1. Discrete variation in fly wings: Wild type wing (left) and two topologically abnormal wings from the Houle Lab, Florida State University.

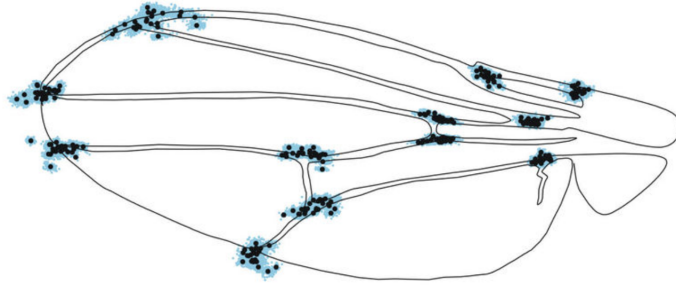


Figure 5.2. Continuous variation in fly wings: Mean locations of landmarks (vein intersections) for 25 species in the sub-family Drosophilinae (black circles) and positions of landmarks for each individual specimen (blue dots) [HM+03].

Persistence is useful for this study because it can measure both types of variation – topological abnormalities and continuous changes in landmark locations – and can naturally fuse these two different types of information together. This is in contrast to other techniques currently used by biologists. See subsections 5.3.1 and 5.3.2 for how persistence detects each type of variation and subsection 5.2.1 for methods of shape quantification other than persistence.

Preliminary progress has been made towards analyzing data from an evolving population of *Drosophila melanogaster* (fruit flies) that the Houle Lab at Florida State University maintains and studies. Existing preprocessing steps created by the Houle

Lab [HM+03] use wing images to produce planar splines, and software written by Beriwal produces piecewise linear approximations of the splines. Preliminary, ad-hoc homology computations and visualizations have been made from simulated data of piecewise linear vein approximations. Details on each of these three processes can be found in section 5.5, and results of the computations on simulated data can be found in section 5.6.

Collaborators

This mathematics is joint work with Surabhi Beriwal, who was an undergraduate at the time, and Ezra Miller. Some works from the same research program are [Ber18; Mil15; Mil17].

Biological collaborators are David Houle and members of the Houle Lab at Florida State University. Their data collection and preprocessing is detailed in [HM+03].

5.1 Project goals

This is a long-term project with goals spanning different timelines. The shortest term goals are about computation, fundamental theoretical concerns, and finding evidence indicating a link between evolution of continuous and topological feature changes in a population of flies. Long-term goals for the project are to find causal links between evolution of continuous and topological changes in morphology, adjust experiments and analysis to study the relationship between genotype and phenotype, and study how development relates to evolution in regards to morphology.

5.1.1 Short-term goals

Our most immediate goal is to finish the wing persistence computational pipeline so we can produce persistence outputs for topologically wild type fly wings (i.e., wings like those on the left in figure 5.1). Once we have access to topologically abnormal wing data (see remark 5.1.1), we have a goal of finding evidence that increased rates of topological abnormalities correlate with extreme continuous evolution (hypothesis 5.2.1). This process may include adjustments to the filtration (see subsection 5.4.1) and analysis of different persistence invariants and metrics to choose the tools that are best able to describe the variation in fly wings.

Some of the steps toward finishing our data pipeline been completed. The Houle lab has collected data in the form of images of wings and has preprocessed the topologically wild type wing images to produce splines (piecewise quadratic polynomials) fitted to the veins of each wing. In order to reduce computation, these splines can be approximated by piecewise linear curves using using software by Beriwal. For more details on the software that provides preprocessing and linear approximation functionalities, see subsections 5.5.1 and 5.5.2.

Some preliminary computations of the persistence filtration and Hilbert function (definition 3.1.3) can be made given piecewise linear approximations of wing veins, but this can only be done for a few data points using ad-hoc methods (ad-hoc methods are described in subsection 5.5.3). More robust and efficient computations using existing multipersistence software are planned.

To make robust persistence computations we plan to use the multipersistence software RIVET [KLW14; LW15] to compute a multipersistence module for each fly wing. The

first attempt at this will be based on discretization, both on the part of the fly wing space and the parameter space. Later attempts may use a finite encoding to reduce the size of the filtration of simplicial complexes that is input into the RIVET software. Details are in remark 5.1.2.

Remark 5.1.1. Some changes need to be made to the fly wing image preprocessing software to record accurate and complete data for topologically abnormal wings. One completed goal of the project is to determine exactly what additional or different information needs to be gathered for topologically abnormal wings and how that information should be recorded. With these new specification in hand, the Houle Lab is now collecting and recording the additional data.

Remark 5.1.2 (Future software). The following is a plan for how to implement the persistence module computations for the fly wings: first, the filtrations have to be constructed from each fly wing as described in subsection 5.3.1. Each filtered piece will essentially be a binary image (with each pixel labeled either 1 or 0). From this filtration of binary images we will produce a filtration of simplicial complexes: each 1-pixel in a binary image gives rise to to a vertex in the corresponding simplicial complex. Vertices are connected by an edge if they are neighbors in the binary image. 2-simplices are added when three vertices are pairwise-neighbors (just as in a Vietoris-Rips complex). No higher simplices are added since these simplicial complexes live in the plane. Once the fly wing data can be represented as a filtration of simplicial complexes, RIVET can compute the corresponding persistence module and some related invariants. Some additional software may need to be produced to extract desired invariants or distances, but as of now our choices of metrics (subsection 5.4.3) can be computed using existing RIVET functionality.

5.1.2 Long-term goals

The long-term biological goals for this research program are to further explore the relationship between evolution, development, genotype, and morphology. This will require setting up new experiments and collecting more data. It will not, however, require much update to the wing persistence computational pipeline. This persistence tool is being designed as a feature extractor for planar graphs, which fly wing veins happen to be. Once the pipeline is created it can be used in any other experiment that has planar graph data and requires measurements of the continuous and topological variation of those graphs.

5.2 Biological context

Current biological work in studying fly wing morphology is much of the time constrained to studying wings with wild type topology (see subsection 5.2.1). The extension to novel topologies is an exciting step forward and allows for the new types of hypotheses to be studied. The hypotheses of interest are in stated in subsection 5.2.2.

5.2.1 Existing techniques for quantifying shape

The Houle Lab’s preprocessing steps for fly wing analysis are incompatible with the project because their models enforce normal topology in their summary of each wing, regardless of whether the wing has normal or abnormal topology. This method has been used to quantify the shape of fly wings by aligning and scaling all samples and comparing the locations and relationships between landmarks/vein intersections

(see Figure 5.2 for examples). The software used to do this – `PREPROCESSWING`– is described in subsection 5.5.1 and [HM+03].

Some scientists compare locations of homologous (in the biological sense!) points (*landmarks*) in wings, e.g. in [BT+07]. This relies on appropriate alignment and sometimes also scaling. Procrustes alignment is can be used, as well as other alignments, e.g. [LH09].

Wing shape can be measured via angular measurements: take two lines on the wing, each connecting a pair of landmarks. There is a baseline difference in angle and angular offset from that is a measure of wing shape. This measure is convenient because 1) scientists don't need to align wings to measure these angles, and 2) the angles are not dependent on size of the wing, which is often something scientists control for by scaling wings to the same size. The first is a property persistence shares and the second only this measurement of angular deviations has, but notice that angular deviation cannot account for a wing that is missing a required landmark (vein intersection) and it does not, necessarily, recognize topological differences in vein topology. [Web90; WE+01]

Sometimes a datum can be missing a landmark; this can happen if samples are incomplete or damaged, or it could happen via mutation or developmental aberration. When a landmark is missing, usually analysis is chosen so that that landmark isn't needed, or the landmark is estimated [AB14]. We are unaware of morphometric methods to deal with topological novelty.

5.2.2 Biological hypotheses

As described in section 5.1 on project goals, this is an ongoing project with short- and long-term goals. Below are some hypotheses we are interested in exploring as well as a description of how persistence output may contribute to supporting a biological hypothesis.

The first hypothesis we will be able to explore involves finding correlation (or independence) between two types of variation.

Hypothesis 5.2.1. *For a population of fruit flies, there is a positive correlation between extreme displacements of landmarks and higher rates of topological novelty.*

Persistent homology measures variation between individual specimens, not the evolution of a population. To study evolution, persistence output can be fed into a statistical framework that analyzes the variation at the population level. If entire populations exhibit the correlation between more extreme landmark displacement and higher rates of topological novelty, then the experiment provides evidence for hypothesis 5.2.1.

The desire to study population-level variation indicates that we should choose persistence invariants and metrics that have well-defined means that both exist and are unique.

Ideally, we would like to explore the hypothesis below.

Hypothesis 5.2.2. *Higher rates of topological variation in fly wing vein morphology occurs as a result of selection towards extreme displacement of landmarks.*

We may be able to provide evidence for hypothesis 5.2.2 because of the way the experiment is set up. Biologists artificially select for extreme displacement of landmarks by removing specimens with less extreme variations in their landmark locations. Any changes in the rate of topological novelty may be caused by that selection towards more extreme landmark displacements.

5.3 Mathematical setting

Below is the translation between the biological problem of analyzing wing venation and the mathematical problem of analyzing persistence modules. Some properties of the data make the persistence computations simpler; they are described at the end of this section.

5.3.1 The two-parameter filtration

Paul Bendich, Joshua Cruz, and Ezra Miller discussed persistent homological summaries of fly wings that were precursors to the 2-parameter filtrations we’re considering now. This two-parameter filtration is inspired at least partially by intersection persistent homology [BH11].

We will now describe the filtration (definition 2.3.13) for our fly wing persistence module. At this point, we will consider the wing as a planar embedding of a graph. The vertices are the vein intersections, or landmarks, and the edges are the veins.

The parameter space for the fly wing filtration (and thus also for the resulting persistence module) is $\mathbb{R}_{\geq 0}^2$ with parameters r and s (i.e., r and s are both nonnegative).

r (= radius) and s (= semi-thickness) are both “thickening” parameters, and each is assigned to a subset of the wing: r is assigned to the vertices/intersections, and s to the edges/veins. For each subset, we consider all the points within a certain distance from that set. That is, we look at all points within distance r from the vertices and all points within distance s from the veins. Our filtered piece is the points within and equaling distance s from the veins minus the points within distance r from the vertices.

$$\begin{aligned} X_{(r,s)} &= \{\vec{x} \in \mathbb{R}^2 \mid d(\text{veins}, x) \leq s, d(\text{vertices}, x) \geq r\} \\ &= \left(\bigcup_{\vec{v} \in \text{veins}} \overline{B_s(\vec{v})} \right) \setminus \left(\bigcup_{\vec{v} \in \text{intersections}} B_r(\vec{v}) \right) \end{aligned}$$

The filtered piece associated to parameters (r, s) is built on top of the wing (i.e. in *wing space*). That means that each filtered piece is a subset of $\mathbb{R}^2$, but note that this is *not* the same as the parameter space $\mathbb{R}_{\geq 0}^2$. In wing space, the distance d is the Euclidean distance.

Figure 5.3 shows the thickening by the two parameters: vertices thickened by r in blue and veins thickened by s in red. The filtered piece associated to (r, s) is the red set minus the blue set. Figure 5.4 shows a simple example of this $\mathbb{R}_{\geq 0}^2$ -filtration, again with thickenings in blue and red. Three filtered pieces are shown, each representing a constant region (definition 2.4.6) of the resulting persistence module.

Remark 5.3.1 (Diagramming over the (r, s) parameter space). In a filtration, the inclusions of filtered pieces are supposed to go in the order of the parameter space poset. For the (r, s) filtration described above, as s increases the filtered pieces add

points but as r increases the filtered pieces lose points. So, when graphing the (r, s) filtration we usually graph in the second quadrant (as opposed to the first quadrant) with r “pointing to the right.” This can be thought of as graphing $(-r, s)$. This way, “up” in this poset is in the expected direction (up and to the right) since as r gets smaller, the filtered piece includes more points.

As a result of this, for every persistence module constructed using this (r, s) filtration, the diagram of that persistence module (usually a map of the Hilbert function, as in figure 5.4) is in the second quadrant of the plane with r increasing toward the left and s increasing toward the top.

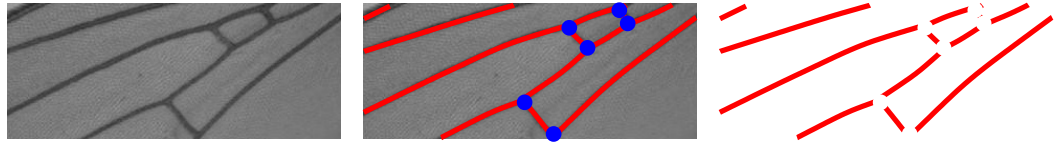


Figure 5.3. From left to right: close-up of a fly wing image; veins (red, semi-thickness s) and vertices (blue, radius r) labeled and thickened according to parameters; the resulting filtered piece.

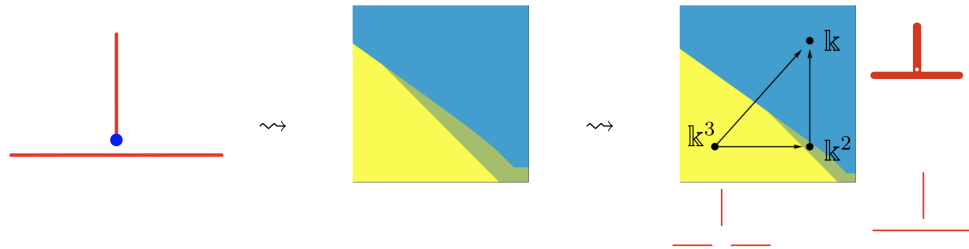


Figure 5.4. From left to right: a toy model fly wing; the wing’s persistence module’s Hilbert function; a finite encoding of the Hilbert function labeled with representatives from the wing’s two-parameter filtration.

5.3.2 Why this filtration?

The reason for this (r, s) filtration is not immediately obvious. We should consider whether a single-parameter filtration would be better. The single parameter could be a thickening parameter for the whole graph or just the vertices. The former setting does keep track of the homology of the wing, which is the most obvious necessity for this application, but the stability of single-persistence [CEH07] causes the resulting topological summary of a wing to give little weight to geometrically small but functionally substantial features. For example, a disconnection that results in the loss of an intersection or the creation of a new but very short vein would cause little change in the resulting persistence module, and as a result it could be difficult to distinguish between persistence module variation due to topological wing variations vs continuous wing variation.

Using a *stratification* – essentially, treating the vertices and veins differently – allows us to pick up on these small but important variations because geometrically small but biologically important wing variations still change the stratification. Changes in stratification cause substantial changes in the persistent homology, so in the end the stratification has the (desirable!) result of destabilizing persistent homology to the types of wing variation we most want to detect. Stratification is discussed further in subsection 5.4.1.

Biologically, it is reasonable to treat vertices and veins differently in this application. The vertices develop differently from, function differently from, and can change the development of veins; for example the genesis of a vertex can cause a vein to form with a kink as it “reaches” to meet and intersect with a different vein. Given the interplay between and biological relevance of both vertices and veins, we consider it

reasonable to treat them differently and assign them different parameters.

This stratified two-parameter filtration for persistence can detect the continuous variation of fly wings alongside the topological variation discussed above. For example, there is a direct effect of changing the angles of pairs of veins joining at a vertex: with fixed s and decreasing r there will be a merge of connected components, as in figure 5.5, at an earlier parameter value when an angle is smaller/more acute. Continuous variation does not change the stratification of the wing and therefore doesn't necessarily cause large changes in the resulting persistence modules, but their effect is still noticeable. An example of this noticeable difference in persistence output is in section 5.6; look for the wings labeled “WT” (wild type) and “CD” (continuous deformation).

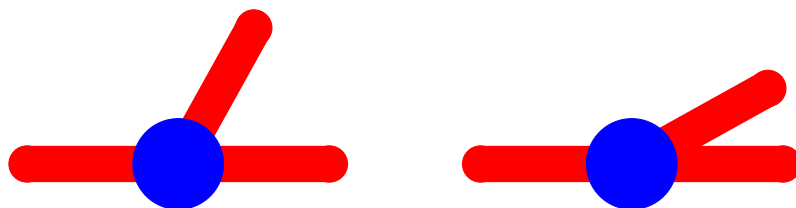


Figure 5.5. The (r, s) filtration detects differences in angles of incoming veins to a vertex: For fixed s and decreasing r , the smaller the angle, the earlier a merge death in H_0 occurs.

5.3.3 Mathematical conveniences for computation

The fact that the fly wing can be considered as embedded in the plane is convenient for mathematical and computational purposes.

- 1) Because the wings can be considered as living in an ambient vector space of dimension 2, only 0^{th} and 1^{st} homology can be non-trivial.

- 2) Alexander duality (theorem 5.3.2) tells us that 1st homology can be computed via a 0th homology computation. As a result the only homology computations that need to be done are 0th homology computations.

Theorem 5.3.2 (Alexander duality). *Let X be a compact, locally contractible, dimension n subspace of S^n . Then for all $q \geq 0$,*

$$\tilde{H}_q(X) \cong \tilde{H}_{n-q-1}(S^n \setminus X)$$

See [MS05, Theorem 5.6] for a simplicial Alexander duality statement.

- 3) 0th homology is easier to work with than higher homology because it deals in connected components. Connected components are easier to compute than higher homologies because they can be computed with standard, existing algorithms used for computer vision. This means that after discretizing the wing space and the parameter space we can use a standard connected components algorithm to compute the homology (see subsection 5.1.1).

5.4 Metrics

To analyze the fly wing persistence modules we need metrics for either the modules or for invariants of the modules. To determine the stability of persistence for this application we also need a metric on the input data, i.e., the wings.

5.4.1 Stability

A stability theorem for a given process is a statement of Lipschitz continuity. For example, the original stability theorem for the bottleneck distance in single-persistence ([CEH07], later improved by [CC+09]) considered the process of taking a topological space X and a function $f : X \rightarrow \mathbb{R}$ and from those constructing a persistence diagram $\text{Dgm}(f)$. The stability statement is that if there are two such functions $f, g : X \rightarrow \mathbb{R}$, then

$$d_{\text{Dgm}}(\text{Dgm}(f), \text{Dgm}(g)) \leq d(f, g)$$

To complete this stability statement we need to define the metrics d_{Dgm} and d .

When dealing with data, measurement, and computation, stability is usually thought of as a necessity because it ensures that if error is introduced into the process – either unintentionally or due to intentional approximations – then the outcome won’t be substantially different from what it would have been without the error.

The fly wing application is a slightly different case, though. Certainly we’d like the small, continuous changes in wings to give small changes in our persistence modules, but we may desire topological changes in the wings to lead to large change in persistence modules because:

- 1) topological differences are most important to us so we don’t want to confuse them with noise or continuous differences;
- 2) the data is nearly certain to have no topological error (i.e., situations where data shows topological variation when there actually isn’t any topological variation in the sample wing, or vice versa); and

- 3) the linear approximation we use on the data does not introduce topological error.

We have chosen a metric for the wings such that continuous variation leads to small changes in the associated persistence modules. The property that allows topological changes to make a sizeable change in the resulting persistence module is the *stratification* of the wing that is used to construct the wing filtration. Choosing a stratification amounts to choosing which vertices are controlled by the radius r parameter. All intersections contribute to the r parameter, but the degree 1 vertices could also contribute. If we choose the appropriate stratifications for the topological differences in fly wings, then we arrange for the desired stability/unstability properties. Figure 5.6 gives an example of different stratifications we could use to analyze a wing that is missing an intersection.

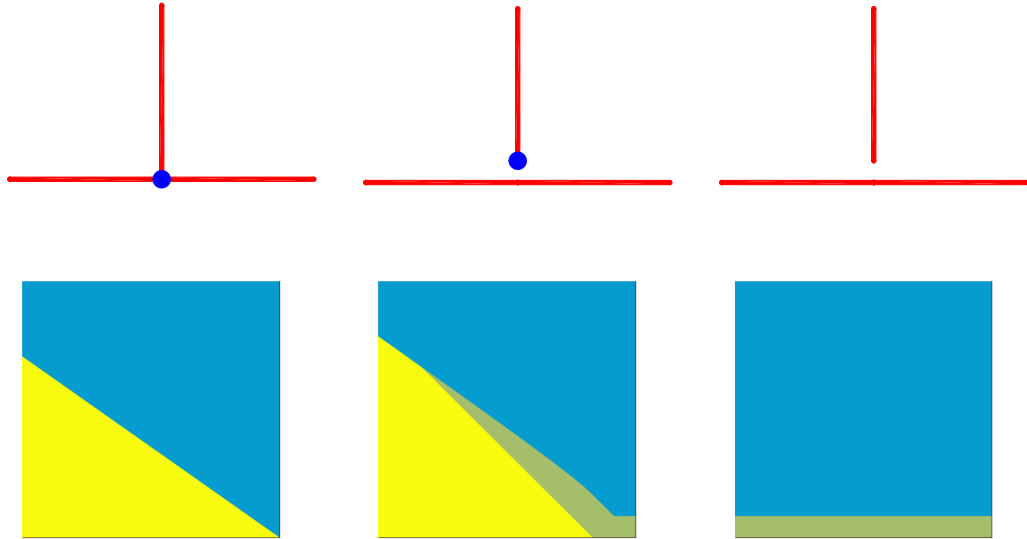


Figure 5.6. Topologically normal wing (left) and two wings missing an intersection. The two abnormal wings show different choices of stratification: one leads to a continuous change in the persistence module while the other leads to a large change. Persistence modules are represented by their Hilbert functions: yellow (lightest) indicates vector spaces of dimension 3; green is dimension 2; blue (darkest) is 1. Figure 5.4 gives a more detailed representation of the middle persistence module.

5.4.2 Metrics on fly wings

A useful distance between two fly wings is the *Hausdorff distance* between their representation as splines. We expect that populations undergoing continuous evolutionary changes will show small distances between parents and offspring but any offspring born with a topological change, depending on how drastic, could be a comparably far distance from its parents.

Definition 5.4.1. The *Hausdorff distance* between subsets A and B of a metric space is

$$\inf\{\epsilon \geq 0 \mid A \subseteq B_\epsilon \text{ and } B \subseteq A_\epsilon\},$$

where X_ϵ is the thickening of X by ϵ .

An algorithm for computing an upper bound on the error of the linear approximation can be found in Beriwal's senior thesis [Ber18], which was co-advised by the author and Ezra Miller.

5.4.3 Metrics on persistence modules

Once the persistence modules are computed there are many different distances and invariants we could use for analysis. Perhaps the simplest starting point is the use the L_p Hilbert distance, as others using multipersistence for applications have [KLW18]. Recall from definition 3.1.3 that the Hilbert function of an $\mathbb{R}^n$ -module M is $\text{Hilb}_M : \mathbb{R}^n \rightarrow \mathbb{N}$

$$\text{Hilb}_M : \mathbf{p} \rightarrow \dim_{\mathbb{k}} M_{\mathbf{p}}$$

and from definition 4.3.26 the L_p Hilbert distance between $\mathbb{R}^n$ -modules M and N is

$$d(M, N) = \|\text{Hilb}_M - \text{Hilb}_N\|_p$$

where $\|\cdot\|_p$ is the usual L_p metric.

On the first pass we will use truncation to make sure the distances between modules are finite. In the future we may use the primary decomposition modification of the L_p Hilbert distance (definition 4.3.31) so as to avoid arbitrary truncation, since we don't know which parts of the module show the most relevant information.

We will likely start by using the L_p distance with $p = 1$ or 2 . A reason to use L_1 is that the L_1 distance does not weight classes all existing at the same parameter values higher than classes existing at different parameter values (see figure 4.2). A reason to use L_2 is the possibility of defining an extended semi-inner product on L_2 space since inner-products allow for the idea of orthogonality and therefore some linear algebraic machine learning techniques like PCA (principal component analysis) and SVM (support vector machines).

With a baseline of analysis done this data set could be used to test the usefulness of other metrics, like L_p metrics on m -rank or metrics with and without primary decomposition modifications.

5.5 Software

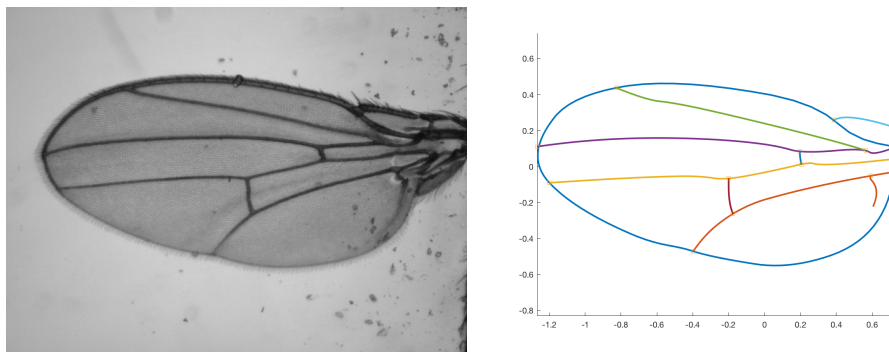
The existing software for this project is described below. PREPROCESSWING was written by member of the Houle Lab [HM+03], LINEARIZEWING was written by a Duke

undergraduate student Surabhi Beriwal under the supervision of this dissertation’s author [Ber18], and the preliminary computations `VISUALIZEFILTRATION` and `ESTIMATEHILBFN` were written by the author of this dissertation. The pipeline for this project is also managed by the author.

5.5.1 Biologist-developed preprocessing

The Houle Lab has developed software to extract vein shape information from images of fly wings with only a small amount of human input. The developers refer to the software as `WINGMACHINE` in their paper [HM+03] but for clarity it will be referred to as `PREPROCESSWING` here.

The input to `PREPROCESSWING` is an image of a fly wing. A human located a small number of landmarks and the rest of the process is automated. The output is a list of splines (parameterized curves), each spline fitted to a vein in the wing. The input and output of `PREPROCESSWING` are shown below.



The automated `PREPROCESSWING` uses the assumption that the input wings are topologically normal. Adding this assumption made the software more accurate and allowed the developers to require less human input to the system. Unfortunately,

that means that there will have to be adjustments made to the software to correctly preprocess the topologically abnormal wings. The number of topologically abnormal wings is a small percentage of the total data set, though, so this could be done “by hand.”

PREPROCESSWING’s processing steps are listed alongside sample data in figure 5.8.

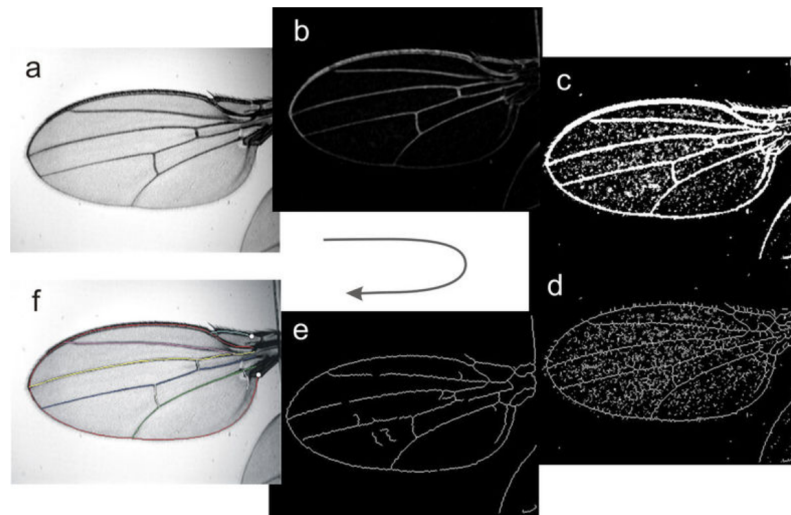


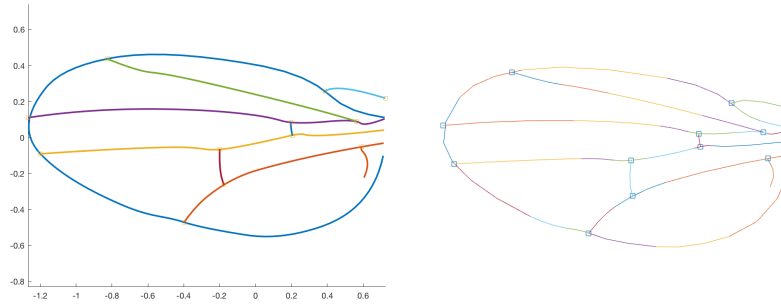
Figure 5.8. PREPROCESSWING image processing of *Drosophila melanogaster* wing: (a) start with raw image (b) invert raw image and minimize background (c) threshold and fill holes (d) reduce features to 1 pixel width (e) remove short features (f) use line intersections to fit model to image and produce final wing image with spline model overlaid [HM+03].

5.5.2 Piecewise-linear fly wing approximations with error bounds

In past experiments, the output of PREPROCESSWING may or may not have been aligned before final analysis of the landmarks. In those analyses all of the feature extraction is done by PREPROCESSWING.

For this project there is more information that needs to be extracted from the wing data. The extraction of topological information involves many distance computations, so to make the computations easier we approximate the wing vein splines by piecewise linear curves using `LINEARIZEWING`.

`LINEARIZEWING` was written by Beriwal under the supervision of this dissertation's author. More details on the software can be found in [Ber18]. Below left are the input data to `LINEARIZEWING` and below right are the vein splines after reparameterization into piecewise quadratic curves. The curves being piecewise quadratic helps with computing the error introduced to the data by the linear approximation.



Each quadratic curve is approximated by a piecewise linear curve, taking into account that arcs with higher curvature should be approximated by shorter line segments. The distance between the quadratic and piecewise linear curves is recorded so that it can be used for an approximation of the error introduced into the data by `LINEARIZEWING`. The introduced error is bounded above by the maximum of these distances.

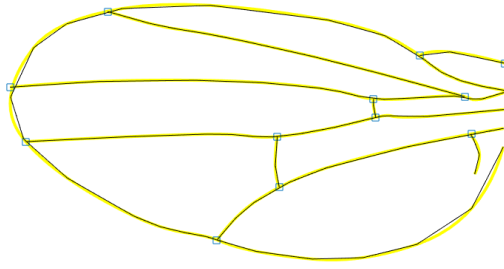


Figure 5.10. Splines (yellow) computed by `PREPROCESSWING`, linear approximation (black) computed by `LINEARIZEWING`, and landmarks (blue squares).

5.5.3 Filtration visualization with connected-component homology estimation

The preliminary persistence computations of wing data have been done with ad-hoc software on simulated data. The software, `VISUALIZEFILTRATION` and `ESTIMATEHILBFN`, output visualizations of the wing filtration and estimations of the persistence module Hilbert function. Some sample visualizations are shown in figures 2.7, 5.4, 5.6 and 5.11.

`ESTIMATEHILBFN`, which estimates the Hilbert function of a wing's persistence module, does not compute the persistence module or even the rank of any of the module's structure maps. The computations are done by identifying connected components in each filtered piece of the fly wing filtration. In these computations both the filtered pieces (in wing space) and the parameters (in the parameter space $\mathbb{R}^2$) are discretized.

For the first analysis of the wing data at scale, we expect to use a faster implementation of similar algorithms.

5.6 Conclusion and preliminary results

Enough of the pipeline has been built (or approximated) that we have determined enough proof of concept to request the full preprocessed data for the topologically abnormal wings. We have communicated to the Houle Lab the extra information needed for each of the topologically abnormal wings to be able to complete our analysis of them.

Some preliminary computations of 0th persistence Hilbert functions computed from simulated wing data can be seen in figure 5.11. The simulated data is of wings that are

- 1) wild type (WT),
- 2) continuously deformed (CD),
- 3) topologically abnormal because they're missing a vertex (MV), and
- 4) topologically abnormal because they have an extra edge (EE).

The topologically abnormal (MV and EE) wings are built from the wild type (WT) wing, so there are both continuous and topological differences between the continuously deformed (CD) wing and the topologically abnormal (MV and EE) wings. Piecewise linear wings with their stratifications are shown beside the approximated Hilbert functions in figure 5.11.

Each pixel in the Hilbert function is the dimension of the homology of a filtered piece of the wing. Recall the coordinates of the Hilbert function: the origin is at the bottom right corner, r increases to the left, and s increases upward. If we think of the parameter space $\mathbb{R}^2$ as a poset, then “up” in the poset is up and to the right and

that is the direction that the modules' structure morphisms (definition 2.3.5) follow. See subsection 5.3.1 for a full description of the filtration and how it relates to the stratification of each wing.

The values of the Hilbert function go from 0 (darkest blue) to 14 (lightest yellow). We previously expected that there may be high values along the antidiagonal $r = s$ due to error introduced by the discretization of the wing filtration, but we are unsure whether that has had a significant effect.

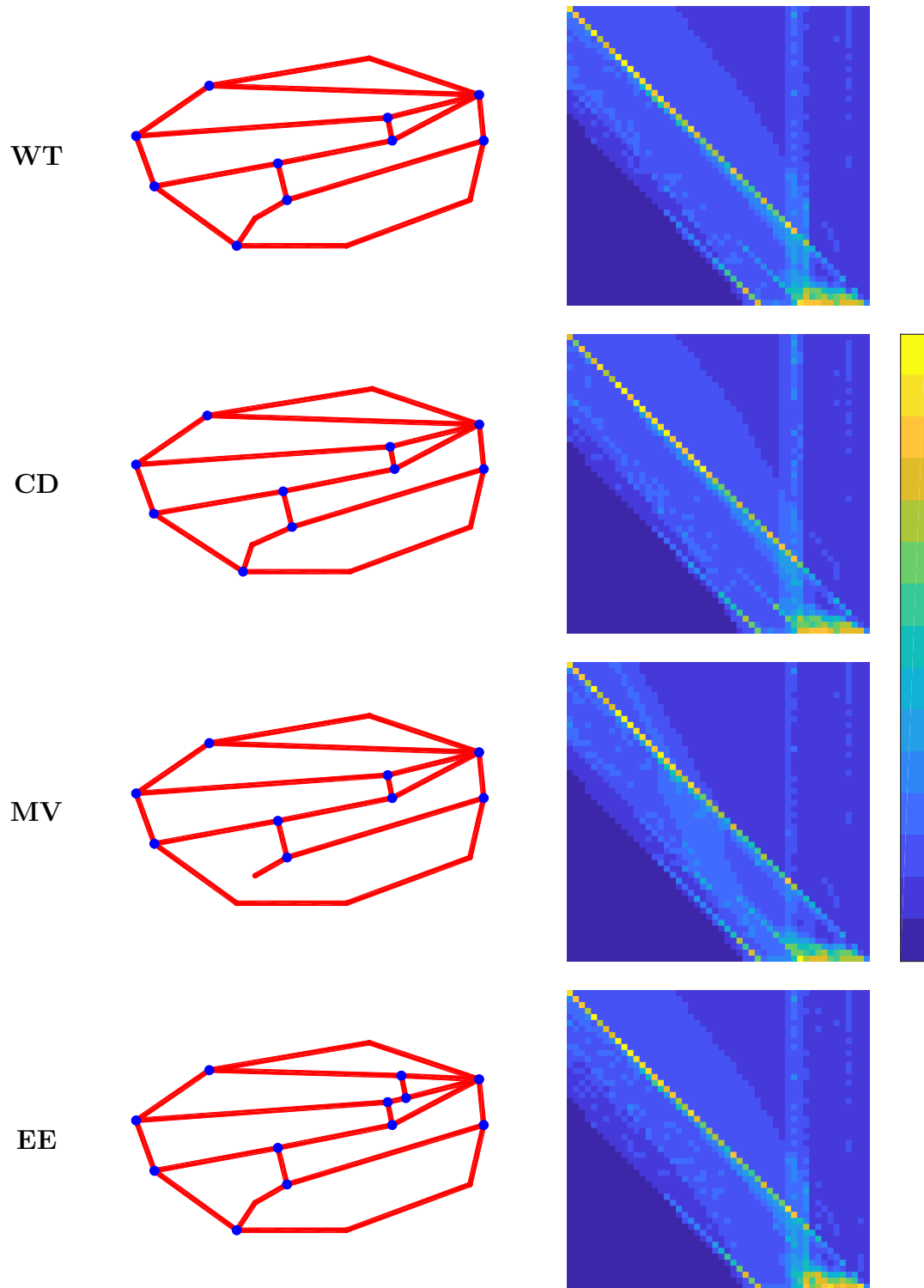


Figure 5.11. Preliminary, approximate Hilbert functions of H_0 persistence fly wing modules. The Hilbert functions (right) show the dimensions of vector spaces in 0^{th} persistence module, colored darkest blue (dimension 0) to lightest yellow (dimension 14).

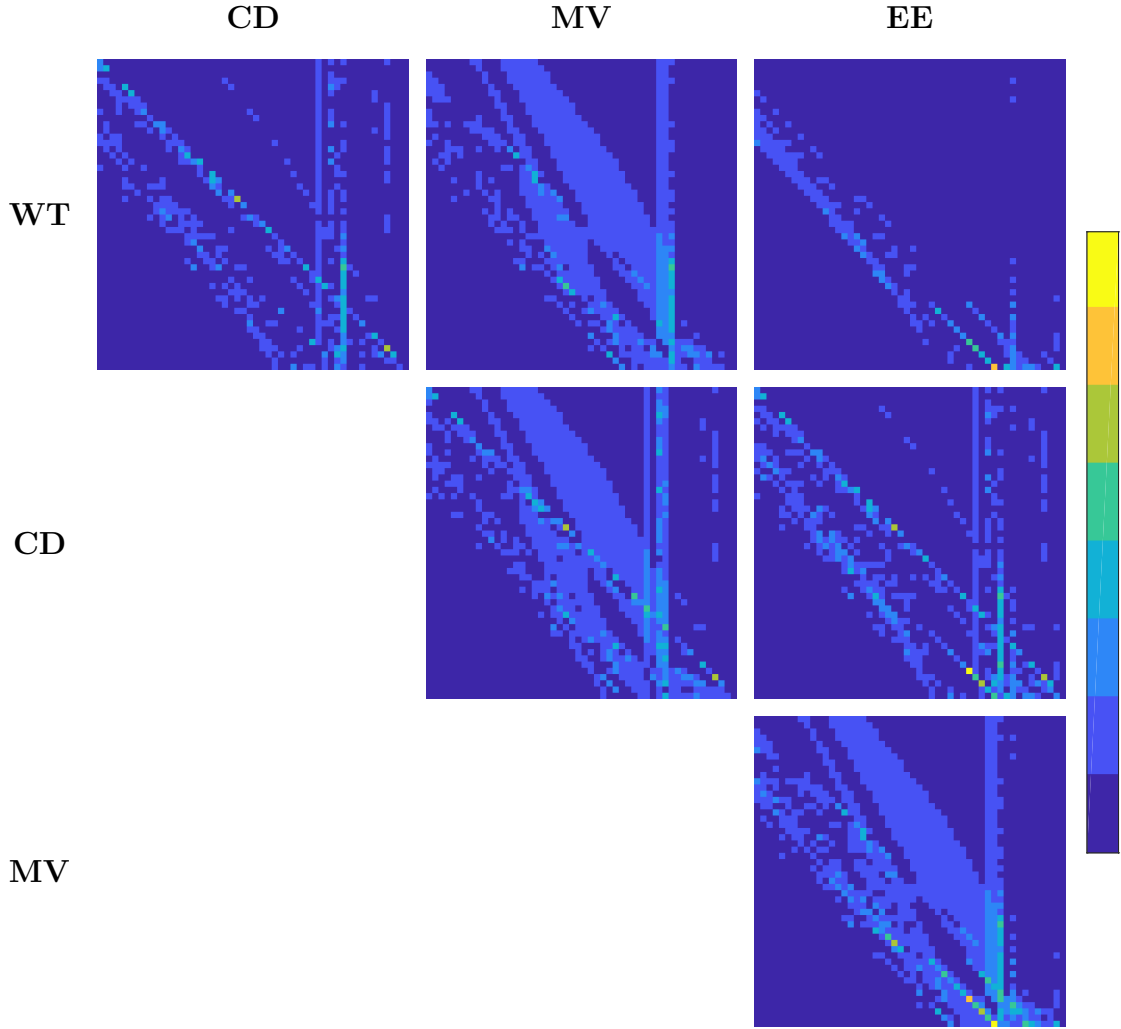


Figure 5.12. Absolute differences between simulated wing Hilbert functions, colored darkest blue (0) to lightest yellow (7).

	WT	CD	MV	EE
WT	0	484	916	226
CD	*	0	1020	628
MV	*	*	0	1054
EE	*	*	*	0

Figure 5.13. Pairwise L_1 -Hilbert distances (definition 4.3.26) between simulated wings.

The pairwise L_1 Hilbert distances (definition 4.3.26) between simulated wings are

shown in figure 5.13. The absolute differences of Hilbert functions, which are used to compute the pairwise distances, are shown in figure 5.12.

The goal of persistence in this application is to detect different types of morphological variation. We hope that persistence is able to differentiate between wild type (WT), continuously deformed (CD), and topologically abnormal (MV, EE) wings, and furthermore that each type of variation moves a fly wing in an independent direction in the space of persistence invariants like the H_0 Hilbert function.

It is easiest to see that persistence has detected the difference between the wing with a missing vertex (MV) and all other wings. This wing is far from the other three wings and in comparison the other wings seems to be in a cluster. It is even easy to see, via the absolute Hilbert function differences in figure 5.12, that the MV should be far from the others. This was expected because the choice of stratification for the MV wing removes a vertex from an area with no other vertices close by. See figure 5.6 for a discussion of why the removal of the vertex should significantly change the Hilbert function.

An interesting result is that the wild type (WT) wing and the topologically abnormal wing with extra cross-vein (EE) are closest to one another, and in fact even closer to one another than the wild type (WT) and continuously deformed (CD) wings.

Consider the placement of the extra cross-vein in figure 5.11. The additional vertices are very close to other vertices, so the difference in stratification won't be very noticeable at high values of r (far to the left in the Hilbert function graphs). We can see that the absolute Hilbert function differences (figure 5.12) between the extra edge (EE) wing and the other wings all have higher values for low-to-medium r and low s along an antidiagonal. The differences are highlighted in figure 5.14.

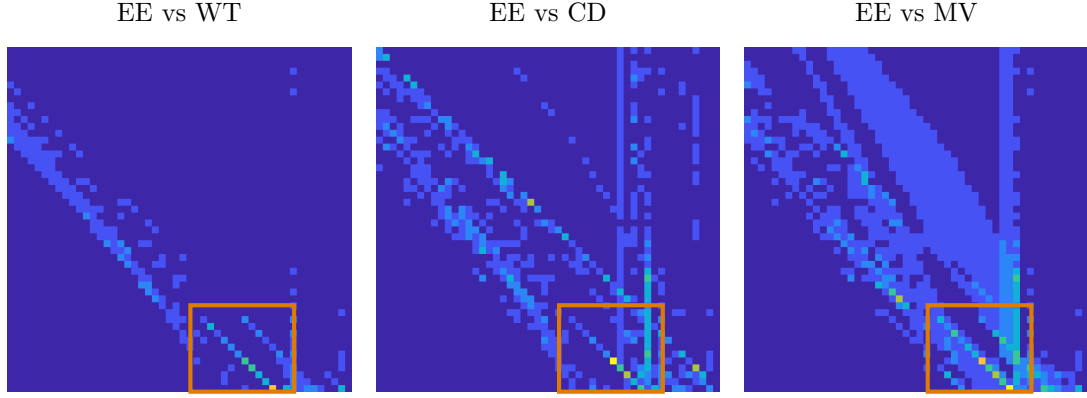


Figure 5.14. Absolute differences of Hilbert functions between a simulated wing with an extra cross-vein (EE) and wild type (WT), continuously deformed (CD), and topologically abnormal with missing vertex (MV) wings. Highlighted is a portion of the Hilbert difference that seems to detect the existence of the extra cross-vein.

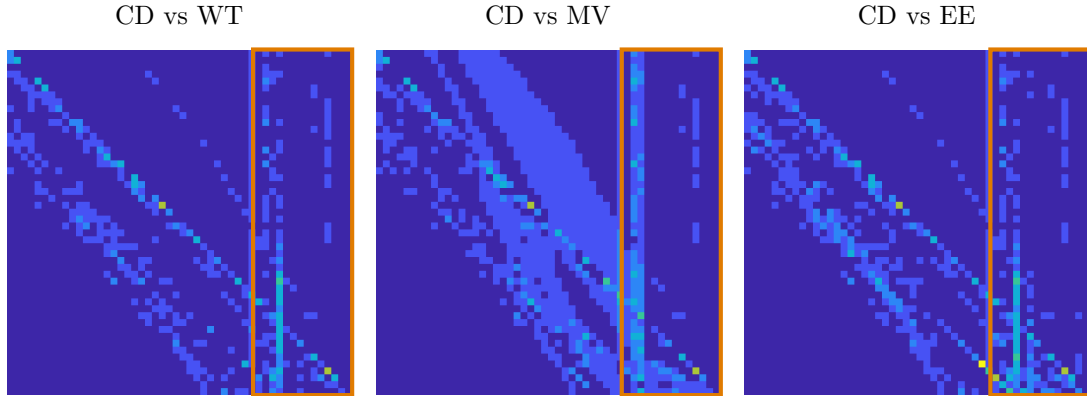


Figure 5.15. Absolute differences of Hilbert functions between a simulated wing that is continuously deformed (CD) and wild type (WT), topologically abnormal with missing vertex (MV), and topologically abnormal with extra edge (EE) wings. Highlighted is a portion of the Hilbert difference that seems to detect the existence of the extra cross-vein.

The differences between the continuously deformed (CD) wing and the other wings seem to be detected in the Hilbert function for low values of r and all values of s (see figure 5.15). The continuous variation is likely picked up most at small r because that is when the thickened vertices are small enough to be distinguished from one

another. At higher values of r , the vertices merge into one contractible mass and give little information about their relative location.

At high values of s the veins join into one large, contractible connected component and for $r < s$, the filtration looks more like (the complement of) a single-persistence filtration for just the vertices, as shown in figure 5.16. The 1st homology of these filtered pieces will correspond to the connected components of the thickened vertices because we are studying the complement of our expected space (see discussion around theorem 5.3.2). This suggests that Hilbert functions of the 1st persistence may give useful information about the amount of continuous deformation from wild type in a wing.

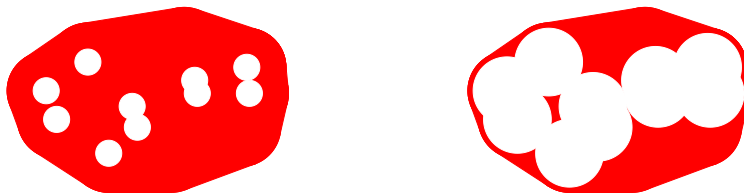


Figure 5.16. For high values of s and $r < s$, the filtered pieces of the fly wing filtration look like (complements of) single-persistence on the vertices. This indicates that 1st homology might be useful for detecting the amount of continuous deformation in a wing.

Since the three types of variation studied in these preliminary computations seem to be detected by different subsets of the domain of the Hilbert function, it is possible that each type variation contributed independently to the change in Hilbert function from that of the wild type (WT) wing. Further analysis about the independence of each variation's effect on the fly wing Hilbert function is future work.

Chapter 6

Conclusion

This dissertation studies invariants and metrics of multiparameter (and generalized) persistent homology so that persistence modules can be more easily compared, used in statistics and machine learning, and applied to shape analysis problems. The theoretical contributions are novel invariants as well as a modification for existing metrics that leads to better distance information between modules with immortal homology classes. The application to evolutionary biology is a new technique for phenotype and morphology quantification that fuses both continuous and topological information into one measurement in a canonical way. Preliminary results from computations on simulated fly wing data show that persistence is detecting the expected features of the data and could be used to give evidence for hypothesis 5.2.1 Future plans involve using contributions from the theoretical portion of the dissertation for final analysis of the fly wing data.

References

- [AB14] Jessica H. Arbour and Caleb M. Brown. “Incomplete specimens in geometric morphometric analyses”. In: *Methods in Ecology and Evolution* 5.1 (Jan. 2014). Ed. by Shinichi Nakagawa, pp. 16–26. DOI: 10.1111/2041-210X.12128. URL: <http://doi.wiley.com/10.1111/2041-210X.12128>.
- [AE+17] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Eric Hanson, Francis Motta, and Lori Ziegelmeier. *Persistence Images: A Stable Vector Representation of Persistent Homology*. Tech. rep. 2017, pp. 1–35. URL: <http://jmlr.org/papers/v18/16-337.html>.
- [Aig07] Martin Aigner. *A Course in Enumeration*. Vol. 238. Graduate Texts in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. ISBN: 978-3-540-39032-9. DOI: 10.1007/978-3-540-39035-0. URL: <http://link.springer.com/10.1007/978-3-540-39035-0>.
- [Bau15] Ulrich Bauer. *Ripser*. 2015. URL: <https://github.com/Ripser/ripser>.
- [BBK18] Håvard Bakke Bjerkevik, Magnus Bakke Botnan, and Michael Kerber. “Computing the interleaving distance is NP-hard”. In: (Nov. 2018). arXiv: 1811.09165. URL: <http://arxiv.org/abs/1811.09165>.
- [BC+08] S. Biasotti, A. Cerri, P. Frosini, D. Giorgi, and C. Landi. “Multidimensional size functions for shape comparison”. In: *Journal of Mathematical Imaging and Vision* 32.2 (Oct. 2008), pp. 161–179. ISSN: 09249907. DOI: 10.1007/s10851-008-0096-z. URL: <http://link.springer.com/10.1007/s10851-008-0096-z>.
- [BC+11] Silvia Biasotti, Andrea Cerri, Patrizio Frosini, and Daniela Giorgi. “A new algorithm for computing the 2-dimensional matching distance between size functions”. In: *Pattern Recognition Letters* 32.14 (2011), pp. 1735–1746. ISSN: 01678655. DOI: 10.1016/j.patrec.2011.07.014.
- [Ber18] Surabhi Beriwal. “Statistical analysis of fruit fly wing vein topology”. Honors Thesis. Duke University, 2018, p. 14. URL: <https://hdl.handle.net/10161/16620>.
- [BGH11] Paul Bendich, Taras Galkovskyi, and John Harer. “Improving homology estimates with random walks”. In: *Inverse Problems* 27.12 (Dec. 2011), p. 124002. ISSN: 02665611. DOI: 10.1088/0266-5611/27/12/124002.

URL: <http://stacks.iop.org/0266-5611/27/i=12/a=124002?key=crossref.2aa7c47be81a140aa611e96959ce02b5>.

- [BGK15] Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. “Persistent Homology for Path Planning in Uncertain Environments”. In: *IEEE Transactions on Robotics* (2015). ISSN: 15523098. DOI: 10.1109/TR0.2015.2412051.
- [BH11] Paul Bendich and John Harer. “Persistent Intersection Homology”. In: *Foundations of Computational Mathematics* 11.3 (June 2011), pp. 305–336. ISSN: 16153375. DOI: 10.1007/s10208-010-9081-1. URL: <http://link.springer.com/10.1007/s10208-010-9081-1>.
- [BH74] James R. Bunch and John E. Hopcroft. “Triangular factorization and inversion by fast matrix multiplication”. In: *Mathematics of Computation* 28.125 (Jan. 1974), pp. 231–236. DOI: 10.1090/S0025-5718-1974-0331751-8. URL: <http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-1974-0331751-8>.
- [BS14] Peter Bubenik and Jonathan A. Scott. “Categorification of Persistent Homology”. In: *Discrete and Computational Geometry* 51.3 (Apr. 2014), pp. 600–627. ISSN: 14320444. DOI: 10.1007/s00454-014-9573-x. URL: <http://link.springer.com/10.1007/s00454-014-9573-x>.
- [BSS15] Peter Bubenik, Vin de Silva, and Jonathan Scott. “Metrics for Generalized Persistence Modules”. In: *Foundations of Computational Mathematics* 15.6 (Dec. 2015), pp. 1501–1531. ISSN: 16153383. DOI: 10.1007/s10208-014-9229-5. arXiv: 1312.3829. URL: <https://doi.org/10.1007/s10208-014-9229-5>.
- [BT+07] O. A. Bubliy, K. S. Tcheslavskaja, A. M. Kulikov, O. E. Lazebny, and V. G. Mitrofanov. “Variation of wing shape in the *Drosophila virilis* species group (Diptera: Drosophilidae)”. In: *Journal of Zoological Systematics and Evolutionary Research* 0.0 (Nov. 2007), 071106232643001–???. ISSN: 0947-5745. DOI: 10.1111/j.1439-0469.2007.00437.x. URL: <http://doi.wiley.com/10.1111/j.1439-0469.2007.00437.x>.
- [Bub15] Peter Bubenik. “Statistical topological data analysis using persistence landscapes”. In: *Journal of Machine Learning Research* 16.1 (2015), pp. 77–102. ISSN: 15337928. arXiv: 1207.6437. URL: <http://arxiv.org/abs/1207.6437> <http://jmlr.org/papers/v16/bubenik15a.html>.
- [CC+09] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. “Proximity of persistence modules and their dia-

- grams”. In: *Proceedings of the 25th annual symposium on Computational geometry - SCG '09*. New York, New York, USA: ACM Press, 2009, p. 237. ISBN: 9781605585017. DOI: 10.1145/1542362.1542407. URL: <http://portal.acm.org/citation.cfm?doid=1542362.1542407>.
- [CCM11] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. “Geometric Inference for Probability Measures”. In: *Foundations of Computational Mathematics* 11.6 (Dec. 2011), pp. 733–751. ISSN: 16153375. DOI: 10.1007/s10208-011-9098-0. URL: <http://link.springer.com/10.1007/s10208-011-9098-0>.
- [CD+13] Andrea Cerri, Barbara Di Fabio, Massimo Ferri, Patrizio Frosini, and Claudia Landi. “Betti numbers in multidimensional persistent homology are stable functions”. In: *Mathematical Methods in the Applied Sciences* 36.12 (Aug. 2013), pp. 1543–1557. ISSN: 01704214. DOI: 10.1002/mma.2704. arXiv: 0908.0064. URL: <http://doi.wiley.com/10.1002/mma.2704>.
- [CDF10] Francesca Cagliari, Barbara Di Fabio, and Massimo Ferri. “One-dimensional reduction of multidimensional persistent homology”. In: *Proceedings of the American Mathematical Society* 138.08 (Aug. 2010), pp. 3003–3003. ISSN: 0002-9939. DOI: 10.1090/s0002-9939-10-10312-8. URL: <http://www.ams.org/jourcgi/jour-getitem?pii=S0002-9939-10-10312-8>.
- [CEF13] Andrea Cerri, Marc Ethier, and Patrizio Frosini. *A study of monodromy in the computation of multidimensional persistence*. 2013. DOI: 10.1007/978-3-642-37067-0-17. URL: http://link.springer.com/10.1007/978-3-642-37067-0%7B%5C_%7D17.
- [CEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. “Stability of Persistence Diagrams”. In: *Discrete & Computational Geometry* 37.1 (Jan. 2007), pp. 103–120. ISSN: 0179-5376. DOI: 10.1007/s00454-006-1276-5. URL: <http://link.springer.com/10.1007/s00454-006-1276-5>.
- [CF10] Andrea Cerri and Patrizio Frosini. “Invariance properties of the multidimensional matching distance in Persistent Topology and Homology”. In: (2010), p. 14. DOI: 10.6092/unibo/amsacta/2765. arXiv: 1004.4753. URL: <http://amsacta.unibo.it/2765/%20http://arxiv.org/abs/1004.4753>.
- [CI+08] Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. “On the local behavior of spaces of natural images”. In: *International*

- Journal of Computer Vision* 76.1 (Jan. 2008), pp. 1–12. ISSN: 09205691. DOI: 10.1007/s11263-007-0056-x. URL: <http://link.springer.com/10.1007/s11263-007-0056-x>.
- [CSV14] Wojciech Chachólski, Martina Scolamiero, and Francesco Vaccarino. “Combinatorial presentation of multidimensional persistent homology”. In: (Sept. 2014). arXiv: 1409.7936. URL: <http://arxiv.org/abs/1409.7936>.
- [CSZ09] Gunnar Carlsson, Gurjeet Singh, and Afra Zomorodian. “Computing multidimensional persistence”. In: *Algorithms and Computation*. Ed. by Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 730–739. ISBN: 978-3-642-10631-6. DOI: 10.1007/978-3-642-10631-6_74. arXiv: 0907.2423.
- [Cur15] Justin Michael Curry. “Topological data analysis and cosheaves”. In: *Japan Journal of Industrial and Applied Mathematics* 32.2 (July 2015), pp. 333–371. ISSN: 1868937X. DOI: 10.1007/s13160-015-0173-9. arXiv: 1411.0613. URL: <http://link.springer.com/10.1007/s13160-015-0173-9>.
- [CW90] Don Coppersmith and Shmuel Winograd. “Matrix multiplication via arithmetic progressions”. In: *Journal of Symbolic Computation* 9.3 (1990), pp. 251–280. ISSN: 07477171. DOI: 10.1016/S0747-7171(08)80013-2. URL: <https://core.ac.uk/download/pdf/82239340.pdf>.
- [CZ09] Gunnar Carlsson and Afra Zomorodian. “The theory of multidimensional persistence”. In: *Discrete and Computational Geometry* (2009). ISSN: 01795376. DOI: 10.1007/s00454-009-9176-0.
- [DW05] Harm Derksen and Jerzy Weyman. “Quiver Representations”. In: *Notices of the AMS* 52.2 (Feb. 2005), pp. 200–206. URL: <http://www.ams.org/notices/200502/fea-weyman.pdf>.
- [Eis95] David Eisenbud. *Commutative algebra with a view toward algebraic geometry*. Springer-Verlag, 1995, p. 785. ISBN: 9780387942698. DOI: 10.1090/S0273-0979-96-00662-3. arXiv: arXiv:1011.1669v3. URL: <http://www.ams.org/journal-getitem?pii=S0273-0979-96-00662-3>.
- [GC17] Oliver Gäfvert and Wojciech Chachólski. “Stable Invariants for Multidimensional Persistence”. In: (Mar. 2017). arXiv: 1703.03632. URL: <http://arxiv.org/abs/1703.03632>.
- [GG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge: Cambridge University Press, 2003. ISBN: 9781139856065. DOI:

10.1017/CB09781139856065. URL: <http://ebooks.cambridge.org/ref/id/CB09781139856065>.

- [Hat02] Allen Hatcher. *Algebraic topology*. Cambridge: Cambridge University Press, 2002, p. 544. ISBN: 0-521-79160-X; 0-521-79540-0.
- [Hen16] Greg Henselman. *Eirene*. 2016. URL: <http://gregoryhenselman.org/eirene/>.
- [HG16] Gregory Henselman and Robert Ghrist. “Matroid Filtrations and Computational Persistent Homology”. In: *ArXiv e-prints* (June 2016). arXiv: 1606.00199 [math.AT].
- [HM+03] David Houle, Jason Mezey, Paul Galpern, and Ashley Carter. “Automated measurement of Drosophila wings”. In: *BMC Evolutionary Biology* 3.1 (Dec. 2003), p. 25. ISSN: 14712148. DOI: 10.1186/1471-2148-3-25. URL: <http://bmcevolbiol.biomedcentral.com/articles/10.1186/1471-2148-3-25>.
- [HO+17] Heather A. Harrington, Nina Otter, Hal Schenck, and Ulrike Tillmann. “Stratifying multiparameter persistent homology”. In: *ArXiv e-prints* (Aug. 2017). arXiv: 1708.07390. URL: <http://arxiv.org/abs/1708.07390>.
- [IMH82] Oscar H Ibarra, Shlomo Moran, and Roger Hui. “A generalization of the fast LUP matrix decomposition algorithm and applications”. In: *Journal of Algorithms* 3.1 (Mar. 1982), pp. 45–56. ISSN: 01966774. DOI: 10.1016/0196-6774(82)90007-4. URL: <https://www.sciencedirect.com/science/article/pii/0196677482900074>.
- [KLW14] Bryn Keller, Michael Lesnick, and Matthew Wright. *RIVET*. 2014. URL: <http://rivet.online/>.
- [KLW18] Bryn Keller, Michael Lesnick, and Ted Willke. *PHoS: Persistent Homology for Virtual Screening*. Aug. 2018. DOI: 10.26434/chemrxiv.6969260.v1.
- [Knu07] Kevin P. Knudson. “A refinement of multi-dimensional persistence”. In: *ArXiv e-prints* (June 2007). arXiv: 0706.2608 [math.AT]. URL: <http://arxiv.org/abs/0706.2608>.
- [LB+17] Peter J. Lawson, J. Quincy Brown, Brittany Terese Fasy, and Carola Wenk. “Architectural features in prostate cancer histology: A case study using persistent homology”. In: *In submission* (2017).
- [Les15] Michael Lesnick. “The Theory of the Interleaving Distance on Multidimensional Persistence Modules”. In: *Foundations of Computational Math-*

- ematics* 15.3 (June 2015), pp. 613–650. DOI: 10.1007/s10208-015-9255-y. arXiv: 1106.5305. URL: <http://link.springer.com/10.1007/s10208-015-9255-y><http://arxiv.org/abs/1106.5305><http://dx.doi.org/10.1007/s10208-015-9255-y>.
- [LH09] Kim van der Linde and David Houle. “Inferring the nature of allometry from geometric data”. In: *Evolutionary Biology* 36.3 (Sept. 2009), pp. 311–322. ISSN: 00713260. DOI: 10.1007/s11692-009-9061-z. URL: <http://link.springer.com/10.1007/s11692-009-9061-z>.
- [LK+12] Hyekyoung Lee, Hyejin Kang, Moo K. Chung, Bung Nyun Kim, and Dong Soo Lee. “Persistent brain network homology from the perspective of dendrogram”. In: *IEEE Transactions on Medical Imaging* 31.12 (Dec. 2012), pp. 2267–2277. ISSN: 02780062. DOI: 10.1109/TMI.2012.2219590. URL: <http://ieeexplore.ieee.org/document/6307875/>.
- [LW15] Michael Lesnick and Matthew Wright. “Interactive Visualization of 2-D Persistence Modules”. In: *ArXiv e-prints* (Dec. 2015). arXiv: 1512.00180. URL: <http://arxiv.org/abs/1512.00180>.
- [Mil15] Ezra Miller. “Fruit flies and moduli: interactions between biology and mathematics”. In: *Notices of the American Math. Society* 62.10 (2015), pp. 1178–1184. DOI: doi:10.1090/noti1290.
- [Mil17] Ezra Miller. “Data structures for real multiparameter persistence modules”. In: *ArXiv e-prints* (Sept. 2017). arXiv: 1709.08155 [math.AT]. URL: <https://arxiv.org/abs/1709.08155>.
- [Mil18] Ezra Miller. “Data structures for real multiparameter persistence modules”. Draft revision of [Mil17] dated 20 September, 2018.
- [MS05] Ezra Miller and Bernd Sturmfels. *Combinatorial Commutative Algebra*. Springer, 2005. ISBN: 0387223568. DOI: 10.4171/OWR/2004/32. arXiv: arXiv:1011.1669v3.
- [MT19] Ezra Miller and Ashleigh Thomas. “Primary decomposition for multipersistence metrics”. In preparation., 2019.
- [Oba18] Ippei Obayashi. “Volume-Optimal Cycle: Tightest Representative Cycle of a Generator in Persistent Homology”. In: *SIAM Journal on Applied Algebra and Geometry* 2.4 (Jan. 2018), pp. 508–534. DOI: 10.1137/17M1159439. URL: <https://epubs.siam.org/doi/10.1137/17M1159439>.
- [RT16] Vanessa Robins and Katharine Turner. “Principal component analysis of persistent homology rank functions with case studies of spatial point pat-

- terns, sphere packing and colloids”. In: *Physica D: Nonlinear Phenomena* 334 (Nov. 2016), pp. 99–117. ISSN: 0167-2789. DOI: 10.1016/J.PHYSD.2016.03.007. arXiv: 1507.01454. URL: <https://www.sciencedirect.com/science/article/pii/S0167278916000476%20https://arxiv.org/pdf/1507.01454.pdf>.
- [SC+17] Martina Scolamiero, Wojciech Chachólski, Anders Lundman, Ryan Ramanujam, and Sebastian Öberg. “Multidimensional Persistence and Noise”. In: *Foundations of Computational Mathematics* 17.6 (Dec. 2017), pp. 1367–1406. ISSN: 16153383. DOI: 10.1007/s10208-016-9323-y. arXiv: 1505.06929. URL: <http://link.springer.com/10.1007/s10208-016-9323-y>.
- [SSV04] B David Saunders, Arne Storjohann, and Gilles Villard. “Matrix rank certification”. In: *Electronic Journal of Linear Algebra* 11 (2004), pp. 16–23. ISSN: 10813810. URL: <http://math.technion.ac.il/iic/ela>.
- [Tho19] Ashleigh Thomas. “Multirank: a multiparameter persistent homology invariant”. In preparation., 2019.
- [Vip18] Oliver Vipond. “Multiparameter Persistence Landscapes”. In: (Dec. 2018). arXiv: 1812.09935. URL: <http://arxiv.org/abs/1812.09935>.
- [WE+01] Kenneth Weber, Robert Eisman, Shawn Higgins, Lisa Morey, April Patty, Michele Tausek, and Zhao-Bang Zeng. “An Analysis of Polygenes Affecting Wing Shape on Chromosome 2 in *Drosophila melanogaster*”. In: *Genetics* 159.3 (2001), pp. 1045–1057. URL: <http://www.genetics.org/content/159/3/1045>.
- [Web90] K E Weber. “Selection on wing allometry in *Drosophila melanogaster*.” In: *Genetics* 126.4 (1990), pp. 975–989.
- [Wie86] Douglas H. Wiedemann. “Solving Sparse Linear Equations Over Finite Fields”. In: *IEEE Transactions on Information Theory* 32.1 (Jan. 1986), pp. 54–62. ISSN: 15579654. DOI: 10.1109/TIT.1986.1057137. URL: <http://ieeexplore.ieee.org/document/1057137/>.
- [Wil12] Virginia Vassilevska Williams. “Multiplying matrices faster than coppersmith-winograd”. In: *Proceedings of the 44th symposium on Theory of Computing*. New York, New York, USA: ACM Press, 2012, p. 887. ISBN: 9781450312455. DOI: 10.1145/2213977.2214056. URL: <http://dl.acm.org/citation.cfm?doid=2213977.2214056>.

Biography

Ashleigh Thomas earned a Bachelor of Science in Electrical Engineering from the University of Pennsylvania in 2013. Part of the way through the program she also took on a second major in math. While at Penn, much of Thomas' work involved robotics.

At the end of her time at Penn Thomas won an NSF Graduate Research Fellowship. She then started graduate school in math at Duke University. While at Duke she participated in an NSF Research Training Group in mathematical biology. Thomas also created and advised 7 short-term group research projects for high school and undergraduate students and co-advised one long-term undergraduate research project with Ezra Miller. Thomas' successful research advising earned her a university-wide award for Excellence in Mentoring.

In spring of 2019 and under the advisement of Ezra Miller, Thomas earned a PhD and Masters of Arts in mathematics from Duke University in Durham, NC. In fall 2019 she will join the faculty at the University of Florida as a Postdoctoral Associate. The position is part of the NSF-Simons Southeast Center for Mathematics and Biology headquartered at the Georgia Institute of Technology.