MuRiT: Efficient Computation of Pathwise Persistence Barcodes in Multi-Filtered Flag Complexes via Vietoris-Rips Transformations

Maximilian Neumann[1], Michael Bleher[2], Lukas Hahn[2], Samuel Braun[3], Holger Obermaier[3], Mehmet Soysal[3], René Caspart[3], Andreas Ott[1,2]

7 July 2022

Abstract

Multi-parameter persistent homology naturally arises in applications of persistent topology to data that come with extra information depending on additional parameters, like for example time series data. We introduce the concept of a Vietoris-Rips transformation, a method that reduces the computation of the one-parameter persistent homology of pathwise subcomplexes in multi-filtered flag complexes to the computation of the Vietoris-Rips persistent homology of certain semimetric spaces. The corresponding pathwise persistence barcodes track persistence features of the ambient multi-filtered complex and can in particular be used to recover the rank invariant in multi-parameter persistent homology. We present `MuRiT`, a scalable algorithm that computes the pathwise persistence barcodes of multi-filtered flag complexes by means of Vietoris-Rips transformations. Moreover, we provide an efficient software implementation of the `MuRiT` algorithm which resorts to `Ripser` for the actual computation of Vietoris-Rips persistence barcodes. To demonstrate the applicability of `MuRiT` to real-world datasets, we establish `MuRiT` as part of our `CoVtRec` pipeline for the surveillance of the convergent evolution of the coronavirus SARS-CoV-2 in the current COVID-19 pandemic.

1. Introduction

Persistent homology is one of the most important tools in computational topology and topological data analysis. It has the capability to detect and explore qualitative features of complex datasets that are encoded in the geometric shape of the dataset and are otherwise hard to extract with traditional methods (see e.g. [EH08, Car09, OPT+17, Ghr07, Wei11, EH10, CdSGO16, Oud15, DW22]). A common approach is Vietoris-Rips persistent homology, which analyzes the geometric shape of metric datasets at varying distance scales. In many applications, however, data points come with extra information that is given in terms of additional attributes and one wishes to leverage this extra information in the topological data analysis. A typical example of this is time series data.

[1]Mathematics Department, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Mathematical Institute, Heidelberg University, Heidelberg, Germany
[3]Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany

Our motivating application in this paper is exactly of this sort—we use persistent homology for the surveillance of emerging adaptive mutations in the evolution of the coronavirus SARS-CoV-2 in the current COVID-19 pandemic [BHPG+21, BHNO22]. Here the dataset consists of coronavirus gene sequences. The use of persistent homology to analyze the evolution of viruses was initiated by Chan, Carlsson and Rabadán [CCR13]. The coronavirus adapts itself to the human host by developing new variants by mutating its genome. In [BHPG+21] we introduced a topological descriptor for the adaptiveness of a given mutation in the genome of the coronavirus that is defined by counting certain one-dimensional cycle representatives in the Vietoris-Rips persistent homology of the gene sequences dataset (see Section 4). Now each coronavirus gene sequence in the dataset is assigned the date at which it was collected from a patient. In this way, the dataset comes with a natural stratification by sampling time, with a bunch of new sequences being added every day.

Ideally, one would like to exploit this additional information and monitor topological signals of adaptation over time in order to tell whether a given mutation is likely to become adaptive in the future. A naive approach is to regard time as an external parameter, and to run the persistence analysis separately for each sub-dataset consisting of all sequences that have been collected up to a given point in time. However, this approach is computationally expensive, as the whole analysis has to be repeated many times. Moreover, classes in persistent homology computed at different time steps will in general not be related with each other. As we will see, all these issues can be resolved by including time as an additional parameter into the persistence analysis itself. The natural setup for this is multi-parameter persistent homology of multi-filtered simplicial complexes introduced by Carlsson and Zomorodian [CZ09, CSZ09]. While it is a challenge to compute multi-parameter persistent homology in general [BL22], it turns out that for our applications in viral evolution one only needs to compute the persistent homology of certain one-filtered subcomplexes in multi-filtered flag complexes.

In the present paper, we address this problem and present MuRiT, a fast and scalable algorithm for the computation of the persistent homology of arbitrary one-filtered subcomplexes of a given multi-filtered flag complex (see Section 3.4). The main idea of the MuRiT algorithm is to apply *Vietoris-Rips transformations* in order to reduce the computation of the persistent homology of one-filtered subcomplexes in a multi-filtered flag complex to the computation of the usual Vietoris-Rips persistent homology of certain semimetric spaces. We will explain Vietoris-Rips transformations in more detail in the next paragraph. The actual computation of the Vietoris-Rips persistent homology of the semimetric space can then be carried out independently with basically any of the presently available software packages [OPT+17], depending on the needs of the particular application one has in mind. However, one has to make sure that the chosen software package is able to handle the Vietoris-Rips persistent homology of *semimetric* spaces that do not necessarily satisfy the triangle inequality.

We provide an efficient software implementation of the MuRiT algorithm at https://github.com/tdalife/murit. In its current form, this implementation is tailored to the case of Vietoris-Rips persistent homology of multi-filtered point cloud datasets, a setup which naturally arises in the Vietoris-Rips persistence analysis of time series data. By default, our implementation of MuRiT resorts to the Ripser software package by Bauer [Bau21b] for the actual computation

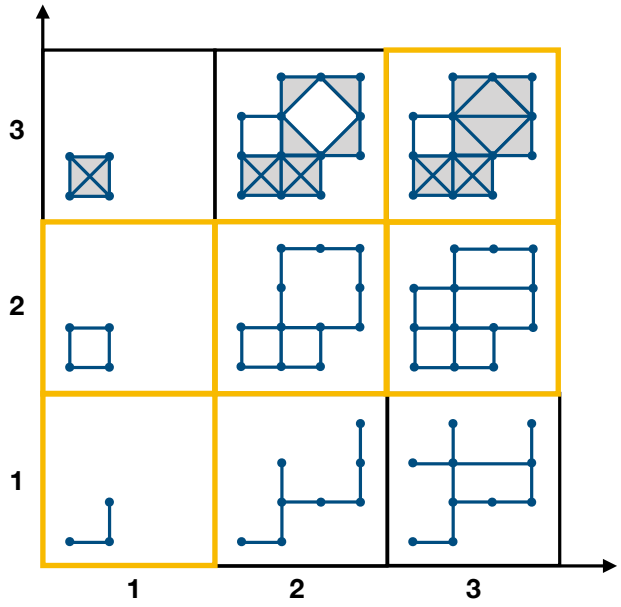

Figure 1: Example of a multi-filtered flag complex. The displayed flag complex X is bi-filtered with three filtration steps in each dimension. The yellow squares mark the one-filtered subcomplex $X_{(1,1)} \subseteq X_{(1,2)} \subseteq X_{(2,2)} \subseteq X_{(3,2)} \subseteq X_{(3,3)}$ of X.

of persistence barcodes. Note at this point that Ripser is able to compute the Vietoris-Rips persistence barcodes also for semimetric spaces that do not necessarily satisfy the triangle inequality [Bau21a]. In this way, MuRiT takes full advantage of the computational power of Ripser, which is among the most efficient implementations for the computation of persistent homology to date [OPT+17]. MuRiT is part of our CoVtRec pipeline for the surveillance of potentially adaptive mutations in the evolution of the coronavirus SARS-CoV-2 in the current COVID-19 pandemic [BHNO22] (see Section 4.3). Thanks to highly optimized algorithms that take advantage of the tree-like structure of the gene sequences dataset [BR22], CoVtRec has the capability to process very large SARS-CoV-2 genomic datasets and easily scales to hundreds of thousands of distinct genomes.

Let us state our main result and outline the basic idea underlying Vietoris-Rips transformations (see Section 3). Assume that X is a finite P-filtered flag complex for some partially ordered set $P = (P, \leq)$, and consider a *(discrete) path* in P that is given by a monotone sequence $\nu = (\nu_1 \leq \nu_2 \leq \nu_3 \leq \dots)$ of elements in P. This gives rise to a one-filtered subcomplex $X_\nu = (X_{\nu_1} \subseteq X_{\nu_2} \subseteq X_{\nu_3} \subseteq \dots)$ of X (see Figure 1). Ideally, for the actual computation of the one-parameter persistent homology of X_ν we would like to resort to any of the currently available efficient algorithms for the computation of Vietoris-Rips persistent homology, like for example Ripser. To that end, we construct a semimetric d on the vertex set $\mathrm{Vert}(X_\nu)$ that encodes the filtration steps of the one-filtered complex X_ν in a suitable way, and define the *Vietoris-Rips transformation* of X_ν as the Vietoris-Rips complex

$$\widehat{\mathrm{VR}}(X_\nu) := \mathrm{VR}(\mathrm{Vert}(X_\nu), d))$$

of the semimetric space $(\mathrm{Vert}(X_\nu), d)$. Then we prove that the one-parameter persistent homology of the Vietoris-Rips transformation $\widehat{\mathrm{VR}}(X_\nu)$ recovers the persistent homology of the original filtration X_ν in the sense that there is an isomorphism

$$H_\ell(X_\nu) \cong H_\ell\big(\widehat{\mathrm{VR}}(X_\nu)\big)$$

of persistence modules in every positive degree $\ell > 0$ (see Theorem 3.1). In the proof we use the fact that the subcomplexes X_{ν_i} are flag complexes. Let us remark that the distance function d will in general not satisfy the triangle inequality, which is why in the definition of the Vietoris-Rips transformation we need to consider Vietoris-Rips complexes of semimetric spaces. We will normally phrase our result in terms of *pathwise barcodes* by saying that in every positive degree $\ell > 0$, the persistence barcode of the complex X along the path ν is computed by

$$\mathcal{B}_\ell(X_\nu) = \mathcal{B}_\ell\big(\widehat{\mathrm{VR}}(X_\nu)\big)$$

in terms of the usual Vietoris-Rips persistence barcode of the Vietoris-Rips transformation $\widehat{\mathrm{VR}}(X_\nu)$ of the one-filtered subcomplex $X_\nu \subseteq X$ (see Section 3.2).

As is shown in [CZ09], there exists no discrete and complete invariant in multi-parameter persistent homology like the persistence barcode known from one-parameter persistence. But there are several approaches to define invariants for multi-persistence, like for example the *rank invariant* introduced in [CZ09], *Hilbert functions* (see e.g. [BL22]), *multi-graded Betti numbers* (see e.g. [MS05]), *signed barcodes* [BOO21] and *fibered barcodes* [LW15, BL22, CFF$^+$13]. Our approach to consider pathwise barcodes of finite multi-filtered flag complexes is reminiscent of the concept of fibered barcodes, where the basic idea is to compute persistence barcodes along affine lines in $\mathbb{R}^n$. In particular, both pathwise and fibered barcodes recover the rank invariant for multi-parameter persistence (see Section 3.3). With `RIVET`, Wright, Lesnick et al. [WLK$^+$20, LW15] provide a software package for working with two-parameter persistent homology, which provides an interactive visualization of the Hilbert function, the bi-graded Betti numbers, and the fibered barcode. Another algorithm specifically designed for the efficient computation of the persistent homology of directed flag complexes is the `Flagser` software package by Lütgehetmann, Govc, Smith and Levi [LGSL19]. A particular feature of `MuRiT` in comparison with `Flagser` is that it does not do the actual computation of persistent homology by itself. In this way, `MuRiT` offers maximum flexibility regarding the choice of software package for the computation of persistent homology. This feature is, for example, indispensable in our application of `MuRiT` to the evolution of the coronavirus, as we need to use a custom version of `Ripser` that is specifically optimized for the efficient localization of cycles in the gene sequences dataset. Another feature of our implementation of `MuRiT` is that it is genuinely designed to deal with pathwise subfiltrations of multi-filtered complexes.

The paper is organized as follows. In Section 2, we fix the notation and recall some basic facts and definitions about the persistent homology of multi-filtered flag complexes. In Section 3, we introduce the notion of a Vietoris-Rips transformation, define pathwise barcodes and relate them with the rank invariant, and present the `MuRiT` algorithm. The final Section 4 discusses an application of the `MuRiT` algorithm to the evolution of the coronavirus.

Acknowledgements. The authors gratefully acknowledge all data contributors, i.e. the Authors and their Originating laboratories responsible for obtaining the specimens, and their Submitting laboratories for generating the genetic sequence and metadata and sharing via the GISAID Initiative [SM17, KGF+21], on which this research is based. An acknowledgement table is accessible online at https://doi.org/10.55876/gis8.220629ug. The authors acknowledge the use of de.NBI Cloud and the support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen and the German Federal Ministry of Education and Research (BMBF) through grant no 031 A535A. They thank M. Hanussek for IT support and early access to VALET [Han21]. The authors further acknowledge support from the Interdisciplinary Center for Scientific Computing at Heidelberg University and the development work of the Scientific Software Center of Heidelberg University carried out by L. Keegan and D. Kempf [KK21]. A.O. acknowledges funding by the Federal Ministry of Education and Research (BMBF) and the Baden-Württemberg Ministry of Science as part of the Excellence Strategy of the German Federal and State Governments (KIT Centers, "Topological Genomics"). A.O. and M.N. acknowledge funding by the Vector Foundation ("Topological Genomics"). L.H. and M.B. were supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster). L.H. thanks the Evangelisches Studienwerk Villigst for their support.

Author Contributions. M.N. developed the concept of Vietoris-Rips tranformations, which grew out of the applications part of M.N.'s Master's thesis under the supervision of A.O.; M.N., M.B. designed and developed the MuRiT algorithm; M.B. designed and implemented the software package MuRiT; M.N., M.B., L.H., A.O. designed and implemented CoVtRec; M.B., L.H., A.O. curated data for CoVtRec; M.N., M.B., L.H., A.O. performed computational analyses; M.N.,M.B., L.H., A.O., S.B., H.O., M.S., R.C. developed and implemented software for CoVtRec; M.B., L.H., A.O. acquired computing resources for CoVtRec; M.N., M.B., L.H., A.O. drafted the manuscript; all authors contributed to the final version of this article.

2. Preliminaries

2.1. Partially Ordered Sets. Let us denote by $\mathbb{N} = \{1, 2, 3, \dots\}$ the set of natural numbers, and write $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For $n \in \mathbb{N}$, we will be working with the following partial order on the n-fold cartesian product $\mathbb{R}^n$ that is induced by the usual total order $\leq$ on the real line $\mathbb{R}$. For any pair of tuples $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ in $\mathbb{R}^n$, we define $a \leq b$ if $a_i \leq b_i$ for all $i \in \{1, \dots, n\}$. The subset $\mathbb{N}^n \subseteq \mathbb{R}^n$ naturally becomes a partially ordered set with the induced partial order $\leq$ inherited from $\mathbb{R}^n$.

A poset $P = (P, \leq)$ is said to have *dimension* n if there exists an order preserving embedding $(P, \leq) \hookrightarrow (\mathbb{R}^m, \leq)$ for $m = n$, but at the same time no such embedding exists for $m < n$.

2.2. Simplicial Complexes and Graphs. We briefly recall some basic facts and definitions about simplicial complexes, and fix some notation and terminology.

An undirected *graph* is a pair $G = (V, E)$ consisting of a set V of *vertices* and a set E of unordered pairs of vertices in V called the *edges*.

An *(abstract) simplicial complex* is a set X of nonempty finite sets such that if σ is an element of X, so is every nonempty subset of σ. The elements of X are called the *simplices* of the complex X. A simplex with $k+1$ elements is called a k*-simplex*, and k is also called its *dimension*. As a particular case of this, 0-simplices in X are also called *vertices* and 1-simplices in X are called *edges*. Any subset of X that is itself a simplicial complex is called a *subcomplex* of X. For every non-negative integer k, the subcomplex $X^{(k)} \subseteq X$ consisting of all simplices in X of dimension at most k is called the k*-skeleton* of X. The 0-skeleton $X^{(0)}$ is also called the *vertex set* $\mathrm{Vert}(X)$ of X, and the complement $\mathrm{Edge}(X) := X^{(1)} \setminus X^{(0)}$ of the vertex set in the 1-skeleton will be called the *edge set* of X.

Consider an undirected graph $G = (V, E)$. A *clique* in the graph G is a finite subset $C \subseteq V$ of the set of vertices such that for any two distinct vertices u and v in C, the unordered pair $\{u, v\}$ formed by these two vertices is contained as an edge in E. We denote by $\mathscr{C}(G)$ the set of all cliques in G. By construction, $\mathscr{C}(G)$ is a simplicial complex and is called the *clique complex* of the graph G.

Let X be a simplicial complex. Observe that the vertex and edge sets of X give rise to an undirected graph

$$G(X) := (\mathrm{Vert}(X), \mathrm{Edge}(X))$$

The complex X is called a *flag complex* if it satisfies the condition $X = \mathscr{C}(G(X))$. In other words, a flag complex is by definition the clique complex of the graph formed by its vertex and edge sets. Then the simplices of the flag complex are precisely the cliques in its vertex set. Note that in this way, every flag complex is completely determined by its 1-skeleton.

2.3. Filtered Sets and Filtered Simplicial Complexes. Let $P = (P, \leq)$ be a poset and X be a set. A P*-filtration* of X is a family of sets $X_\bullet = (X_p)_{p \in P}$ satisfying the following conditions:

(i) $X_p \subseteq X$ is a subset for every $p \in P$.

(ii) $X_p \subseteq X_q$ for all $p, q \in P$ with $p \leq q$.

(iii) $\bigcup_{p \in P} X_p = X$.

A set X is called P-filtered if it admits a P-filtration. If P is n-dimensional, X is called n*-filtered*. X is called *multi-filtered* if X is n-filtered for some $n \geq 2$.

A simplicial complex X is called P*-filtered* if X is a P-filtered set and $X_p \subseteq X$ is a subcomplex for every $p \in P$. A P-filtered simplicial complex X is called a P*-filtered flag complex* if X is a flag complex and the subcomplexes X_p are flag complexes for all $p \in P$.

2.4. Vietoris-Rips Complexes of Semimetric Spaces. In the literature, Vietoris-Rips complexes are normally defined for metric spaces. It is key to our approach in this paper to consider Vietoris-Rips complexes for a larger class of spaces equipped with a more general notion of distance function that is not required to satisfy the triangle inequality. Let S be a set and $[0,\infty] = \mathbb{R}_{\geq 0} \cup \{\infty\}$, where ∞ henceforth denotes $+\infty$ for short. A function $d\colon S \times S \to [0,\infty]$ is called a *semimetric* if it satisfies the following two axioms:

(i) $d(x,y) = d(y,x)$ for all $x, y \in S$.

(ii) $d(x,y) = 0$ if and only if $x = y$, for all $x, y \in S$.

The pair (S,d) is called a *semimetric space*. The function d will also be called the *distance function* of the semimetric space (S,d). Note that a semimetric space is not required to satisfy the triangle inequality, and that we allow the distance function to take the value ∞.

Let (S,d) be a semimetric space. For every $r \in [0,\infty)$, the *Vietoris-Rips complex of* (S,d) *at scale* r is the abstract simplicial complex defined by

$$\mathrm{VR}_r(S,d) := \big\{\sigma \subseteq S \,\big|\, \emptyset \neq \sigma \text{ finite and } d(x,y) \leq r \text{ for all } x,y \in \sigma\big\}.$$

Let us remark that this definition makes sense also if the distance function d is not required to satisfy the triangle inequality. Note moreover that $\mathrm{VR}_r(S,d)$ is in fact a flag complex. Its simplices are precisely all finite non-empty subsets of the set S whose points have pairwise distance at most r. We will also consider the simplicial complex

$$\mathrm{VR}(S,d) := \big\{\sigma \subseteq \mathrm{Vert}(X) \,\big|\, \emptyset \neq \sigma \text{ finite and } d(x,y) < \infty \text{ for all } x,y \in \sigma\big\}.$$

It is a flag complex and we will refer to it as the *Vietoris-Rips complex* of the semimetric space (S,d). Note that $\mathrm{VR}(S,d)$ becomes a $[0,\infty)$-filtered flag complex with Vietoris-Rips filtration $\mathrm{VR}_\bullet(S,d) = (\mathrm{VR}_r(S,d))_{r\in[0,\infty)}$.

2.5. Persistence Modules. Let $P = (P,\leq)$ be a poset, and fix a coefficient field $\mathbb{F}$. In later computations we will normally choose $\mathbb{F} = \mathbb{F}_p$ to be a finite field of prime order. A *persistence module* over P is a functor

$$M : P \to \mathbf{Vec}_{\mathbb{F}}, \quad p \mapsto M(p)$$

from P into the category $\mathbf{Vec}_{\mathbb{F}}$ of vector spaces over the field $\mathbb{F}$. It assigns to every pair $p, q \in P$ with $p \leq q$ an $\mathbb{F}$-linear map denoted by

$$M(p \leq q) : M(p) \to M(q).$$

If P is n-dimensional, then M is also called an *n-parameter persistence module*. M is called a *multi-parameter persistence module* if M is an n-parameter persistence module for some $n \geq 2$. A persistence module M over P is called *pointwise finite dimensional* if $M(p)$ is a finite dimensional $\mathbb{F}$-vector space for all $p \in P$.

Let now X be a finite P-filtered simplicial complex. Then for every non-negative integer $\ell \geq 0$, the assignment

$$H_\ell(X) : P \to \mathbf{Vec}_{\mathbb{F}}, \quad p \mapsto H_\ell(X_p)$$

defines a pointwise finite dimensional persistence module over P, where $H_\ell(X_p) = H_\ell(X_p; \mathbb{F})$ denotes the ℓ-th simplicial homology of X_p with coefficients in the field $\mathbb{F}$.

3. Main Results

3.1. Vietoris-Rips Transformations. We start with a construction that assigns a finite semimetric space to any finite $\mathbb{N}$-filtered flag complex X. For this, we turn the vertex set of X into a semimetric space by explicitly constructing a semimetric d on $\mathrm{Vert}(X)$ in the following way. Let $x, y \in \mathrm{Vert}(X)$ be any pair of vertices. If $x \neq y$, we set

$$d(x,y) := \begin{cases} \min\{i \mid \{x,y\} \in X_i\} \text{ if } \{x,y\} \in \mathrm{Edge}(X), \\ \infty \text{ otherwise,} \end{cases} \tag{1}$$

while if $x = y$, we set $d(x,y) := 0$. So the distance $d(x,y)$ between any two distinct vertices x, y is given by the smallest filtration step at which the edge $\{x,y\}$ enters into the filtration $X_\bullet$, and it is assigned the value ∞ if $\{x,y\}$ is not an edge in X. Note that since $\mathbb{N}$ does not contain the number zero, we have $d(x,y) = 0$ if and only if $x = y$. We remark that the distance function d only defines a semimetric as it will in general not satisfy the triangle inequality. With this understood, the *Vietoris-Rips transformation* of X is defined as the $\mathbb{N}$-filtered Vietoris-Rips complex

$$\widehat{\mathrm{VR}}(X) := \mathrm{VR}(\mathrm{Vert}(X), d)$$

with filtration $\widehat{\mathrm{VR}}_\bullet(X) = (\mathrm{VR}_i(\mathrm{Vert}(X), d))_{i \in \mathbb{N}}$. Our main result now states that the persistent homology in degree greater than zero of the $\mathbb{N}$-filtered flag complex X can be computed in terms of the usual one-parameter persistent homology of its Vietoris-Rips transformation.

Theorem 3.1. *Let X be a finite $\mathbb{N}$-filtered flag complex. Then in every positive degree $\ell > 0$, the Vietoris-Rips transformation induces an isomorphism*

$$H_\ell(X) \cong H_\ell\big(\widehat{\mathrm{VR}}(X)\big) \tag{2}$$

of persistence modules over $\mathbb{N}$.

Let us remark that the isomorphism (2) will in general not hold in degree $\ell = 0$. This is because the filtration $\widehat{\mathrm{VR}}_\bullet(X)$ is defined on the vertex set $\mathrm{Vert}(X)$ of the whole complex, while the vertex set $\mathrm{Vert}(X_i)$ can be a proper subset of $\mathrm{Vert}(X)$.

Proof of the theorem. By construction of the semimetric d in (1), for every $i \in \mathbb{N}$ we have an identity

$$X_i^{(1)} = \mathrm{VR}_i(\mathrm{Vert}(X), d)^{(1)} \setminus (\mathrm{Vert}(X) \setminus \mathrm{Vert}(X_i))$$

of one-dimensional simplicial complexes, where on the right-hand side we need to remove all vertices in X that are not contained in X_i. Since both X_i and $\mathrm{VR}_i(\mathrm{Vert}(X), d)$ are flag complexes, the above identity extends to an identity

$$X_i = \mathrm{VR}_i(\mathrm{Vert}(X), d) \setminus (\mathrm{Vert}(X) \setminus \mathrm{Vert}(X_i)).$$

In particular, this identity means that the complexes X_i and $\mathrm{VR}_i(\mathrm{Vert}(X), d)$ consist of the same k-simplices for $k \geq 1$. Hence we immediately obtain the claimed isomorphism of persistence modules in all positive degrees $\ell > 0$. □

3.2. Pathwise Persistence Barcodes. A non-empty subset $I \subseteq \mathbb{N}$ is called an *interval* if $r \leq s \leq t$ with $r, t \in I$ implies $s \in I$. We define a persistence module $\mathbb{F}_I$ over $\mathbb{N}$ as follows:

$$\mathbb{F}_I(t) = \begin{cases} \mathbb{F} \text{ if } t \in I, \\ 0 \text{ otherwise,} \end{cases}$$

and $\mathbb{F}_I(s \leq t)$ is the identity map for all $s, t \in I$ with $s \leq t$ and the zero map otherwise. $\mathbb{F}_I$ is also called an *interval module*.

Let M be a persistence module over $\mathbb{N}$ of *finite type*, i.e. M is pointwise finite dimensional and there exists some $N \in \mathbb{N}$ such that $M(N \leq m)$ is an isomorphism of $\mathbb{F}$-vector spaces for all $m \in \mathbb{N}$ with $N \leq m$. The structure theorem for one-parameter persistence modules then states that M admits a decomposition

$$M \cong \bigoplus_{j=1}^{m} \mathbb{F}_{I_j}$$

for a finite family $\mathcal{B}(M) = (I_1, \ldots, I_m)$ of intervals which is uniquely determined up to the ordering of the intervals. This family $\mathcal{B}(M)$ is called the *(persistence) barcode* of M. The intervals in the persistence barcode are also called *bars*.

If $M = H_\ell(X)$ is the persistent homology in fixed degree $\ell \geq 0$ of some finite $\mathbb{N}$-filtered simplicial complex X, then $H_\ell(X)$ is of finite type and we denote its persistence barcode by $\mathcal{B}_\ell(X)$. The persistence barcode $\mathcal{B}_\ell(X)$ encodes the persistent homology of X in degree ℓ. The starting point of each bar in the persistence barcode corresponds to the birth of a homology feature, while its endpoint, if it exists, marks the death of the feature. The material about persistence barcodes summarized here is standard and can for example be found in [ZC05] or [CZCG05, §5.2].

Let $P = (P, \leq)$ be a poset. A monotone sequence $\nu = (\nu_1 \leq \nu_2 \leq \nu_3 \leq \dots)$ of elements in P is also called a *(discrete) path* in P. The sequence ν *stabilizes* if there exists some $m \in \mathbb{N}$ such that $\nu_i = \nu_m$ for all $i \geq m$. In this case, we use the notation $\nu = (\nu_1 \leq \cdots \leq \nu_m)$. Let us now consider a finite P-filtered flag complex X. Any path $\nu = (\nu_i)_{i \in \mathbb{N}}$ in P gives rise to an $\mathbb{N}$-filtered subcomplex $X_\nu = \bigcup_{i \in \mathbb{N}} X_{\nu_i}$ of X with filtration $(X_{\nu_i})_{i \in \mathbb{N}}$. As an immediate consequence of Theorem 3.1, in every positive degree $\ell > 0$, the persistence barcode of X_ν may be computed in

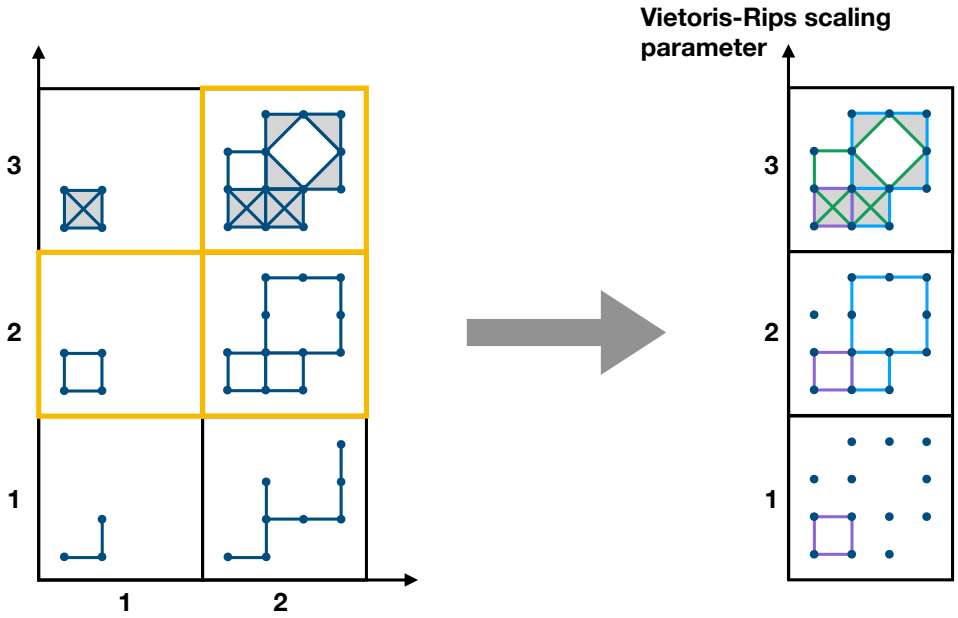

Figure 2: Example of a Vietoris-Rips transformation. On the left, we see a bi-filtered flag complex X. The yellow squares mark the one-filtered subcomplex $X_{(1,2)} \subseteq X_{(2,2)} \subseteq X_{(2,3)} = X_\nu$ defined by the path $\nu = ((1,2) \leq (2,2) \leq (2,3))$. The Vietoris-Rips transformation $\widehat{\mathrm{VR}}(X_\nu)$ of this subcomplex is the Vietoris-Rips complex shown on the right. The coloring of the edges indicates at which scale an edge enters into the Vietoris-Rips filtration.

terms of a Vietoris-Rips persistence barcode of the Vietoris-Rips transformation as

$$\mathcal{B}_\ell(X_\nu) = \mathcal{B}_\ell\big(\widehat{\mathrm{VR}}(X_\nu)\big).$$

We refer to $\mathcal{B}_\ell(X_\nu)$ as the *persistence barcode of X along the path ν*. An instructive example of a Vietoris-Rips transformation is shown in Figure 2.

Our result demonstrates the usefulness of Vietoris-Rips transformations in practice. In fact, it reduces the computation of pathwise persistence barcodes in a finite multi-filtered flag complex to the computation of the persistence barcode of certain Vietoris-Rips filtrations. In Section 3.4, we will present the `MuRiT` algorithm, a software implementation of the Vietoris-Rips transformation for the efficient computation of pathwise persistence barcodes of multi-filtered flag complexes.

3.3. Pathwise Persistence Barcodes and the Rank Invariant. Let X be a finite P-filtered flag complex. Pathwise persistence barcodes are closely related to the rank invariant of $H_\ell(X)$ introduced by Carlsson and Zomorodian [CZ09]. Let $P^2_\Delta = \{(v,w) \in P^2 \mid v \leq w\}$. Now the rank invariant of $H_\ell(X)$ is given as the assignment

$$P^2_\Delta \to \mathbb{N}_0, \quad (v,w) \mapsto \mathrm{rank}(H_\ell(X_v) \to H_\ell(X_w)).$$

It captures important persistence features of the multiparameter persistence module $H_\ell(X)$ and, as Carlsson and Zomorodian observerd, it is equivalent to the persistence barcode in the case of one-parameter persistence. We can recover the rank invariant of $H_\ell(X)$ by computing the persistence barcode along the path $(\nu_1 \leq \nu_2)$ for every pair $(\nu_1, \nu_2) \in P^2_\Delta$.

3.4. The MuRiT Algorithm for Multi-Filtered Flag Complexes. Based on our theoretical considerations in the previous subsections, we now introduce the MuRiT algorithm, displayed in Algorithm 1. It is designed for the computation of pathwise persistence barcodes in positive homology degree of finite multi-filtered flag complexes via Vietoris-Rips transformations. The MuRiT algorithm firstly efficiently computes the Vietoris-Rips transformation, and secondly uses Ripser to compute the persistence barcodes of this Vietoris-Rips transformation.

Our setup for MuRiT will be a finite P-filtered flag complex X with filtration $X_\bullet = (X_p)_{p\in P}$ for some finite n-dimensional subposet $P \subseteq \mathbb{R}^n$. This ensures that MuRiT will be applicable to a large class of real-world data, like for example time series data. Instead of encoding the full complex X, it will be sufficient to specify a finite *edge entry annotation list*

$$L : \mathrm{Edge}(X) \to \mathcal{P}(P)$$

which takes values in the power set $\mathcal{P}(P)$ of P and records, for every edge $\{x, y\} \in \mathrm{Edge}(X)$, the minimal filtration steps $p \in P$ at which this edge enters into the filtration:

$$L(\{x, y\}) := \min\{p \in P \mid \{x, y\} \in X_p\} \subseteq P$$

Recall at this point that minima of subsets of posets are in general not unique. Lastly, we need to specify a path $\nu = (\nu_1 \leq \cdots \leq \nu_m)$ in P, which defines the one-filtered subcomplex $X_\nu \subseteq X$ we would like to analyze.

From this input data, MuRiT first computes the lower triangular distance matrix D of the following semimetric d on the vertex set $\mathrm{Vert}(X)$: the restriction of d to the vertex set $\mathrm{Vert}(X_\nu)$ coincides with the semimetric (1) associated with the Vietoris-Rips transformation $\widehat{\mathrm{VR}}(X_\nu)$, while for any pair $x, y \in \mathrm{Vert}(X) \setminus \mathrm{Vert}(X_\nu)$, we set $d(x, y) := \infty$ if $x \neq y$ and $d(x, y) = 0$ if $x = y$. This makes the algorithm more user friendly—the user only has to encode the complex X once by specifying the annotation list L. After that, they only have to define the paths in P along which they want to compute persistence barcodes.

To fix the notation, we denote the vertices in X by $\mathrm{Vert}(X) = \{x_1, \ldots, x_N\}$. Then D is given by $D_{ij} = d(x_i, x_j)$ with $i > j$. In order to determine the distance D_{ij}, MuRiT calculates the unique minimal intersection of the upper set of $L(\{x_i, x_j\})$ with the given path ν in P. Recall that for a subset $Q \subseteq P$, the upper set of Q in P is defined as the set of all $p \in P$ such that $q \leq p$ for some $q \in Q$. In a second step, MuRiT passes the distance matrix D to Ripser for the actual computation of the persistence barcodes $\mathcal{B}_\bullet(X_\nu) := (\mathcal{B}_\ell(X_\nu))_{\ell\geq 1}$ in positive homology degree. Note at this point that the distance matrix D will in general not satisfy the triangle inequality. But this is not a problem as Ripser can handle distance matrices of semimetric spaces and in particular does not require the matrix D to satisfy the triangle inequality [Bau21a].

Algorithm 1 MuRiT algorithm

Input:

Vertex Set $\mathrm{Vert}(X) = \{x_1, \ldots, x_N\}$
Edge Entry Annotation List $L : \mathrm{Edge}(X) \to \mathcal{P}(P)$
Path $\nu = (\nu_1 \leq \cdots \leq \nu_m)$ in P

Output:

Persistence barcodes $\mathcal{B}_\bullet(X_\nu)$

for every x_i, x_j in $\mathrm{Vert}(X)$ with $i > j$ **parallel do**
  $D(x_i, x_j) := \infty$
  for every step ν_k in the path ν **do**
    for every filtration step p in the edge entry annotation $L(\{x_i, x_j\})$ **do**
      if $p \leq \nu_k$ **then** ▷ check if the edge $\{x_i, x_j\}$ is contained in X_{ν_k}
        $D(x_i, x_j) = k$
        goto `bottom`
      end if
    end for
  end for
  `bottom`
end parallel do
$\mathcal{B}_\bullet(X_\nu) :=$ Ripser(D)
return $\mathcal{B}_\bullet(X_\nu)$

A common way in which multi-filtered flag complexes naturally arise in applications is in the Vietoris-Rips persistence analysis of multi-filtered metric datasets that come with extra structure given by additional parameters. A typical example of this is time series data in the evolution of the coronavirus (see Section 4). To formalize this, let (S, h) be a finite semimetric space equipped with a filtration $S_\bullet = (S_t)_{t \in T}$ for some finite n-dimensional subposet $T \subseteq \mathbb{R}^n$. For example, in the case of time series data we could have a totally ordered subset $T = \{t_1 \leq \ldots \leq t_m\} \subseteq \mathbb{R}$ that specifies the time steps. We may then consider the Vietoris-Rips complex $\mathrm{VR}_r(S_t, h)$ for each filtration step $t \in T$ and $r \in h(S)$, where $h(S)$ denotes the set of pairwise distances $h(x, y)$ of elements $x, y \in S$. This gives rise to a multi-filtered flag complex in the following way. Consider the poset

$$P := T \times h(S) \subseteq \mathbb{R}^{n+1}.$$

Then $\mathrm{VR}(S, h)$ naturally becomes a P-filtered flag complex with filtration

$$\mathrm{VR}_\bullet(S_\bullet, h) = (\mathrm{VR}_r(S_t, h))_{(t,r) \in P}$$

As a result, we may use MuRiT to efficiently investigate the multi-parameter persistent homology of the multi-filtered flag complex $\mathrm{VR}(S, h)$ by computing pathwise persistence barcodes via Vietoris-Rips transformations.

In order to be able to run the MuRiT Algorithm 1, we first need to prepare the input data, which will be done with Algorithm 2. We denote the points in the dataset by $S = \{x_1, \ldots, x_N\}$.

Moreover, we denote by H the lower triangular distance matrix of the semimetric space (S, h) given by $H_{ij} = h(x_i, x_j)$ with $i > j$. Algorithm 2 takes as input this lower triangular distance matrix H, a path ν in P of the form $\nu = ((t_1, r_1) \leq \cdots \leq (t_m, r_m))$, and a *point entry annotation list* $K : S \to \mathcal{P}(T)$ where for every point $x \in S$

$$K(x) := \min\{t \in T \mid x \in S_t\} \subseteq T.$$

The output of Algorithm 2 is the edge entry annotation list $L_\nu : \text{Edge}(\text{VR}(S, h)_\nu) \to \mathcal{P}(P)$ of the one-filtered subcomplex $\text{VR}(S, h)_\nu \subseteq \text{VR}(S, h)$ given by

$$L_\nu(\{x, y\}) = \min\{(t_i, r_i) \in \nu \mid \{x, y\} \in \text{VR}_{r_i}(S_{t_i}, h)\} \subseteq P.$$

We may then pass as input for Algorithm 1 the one-filtered subcomplex $\text{VR}(S, h)_\nu \subseteq \text{VR}(S, h)$ with vertex set $S_{t_m} \subseteq S$, together with the annotation list L_ν.

Algorithm 2 Preparation of Multi-Filtered Data for MuRiT

Input:

Lower Triangular Distance Matrix H
Point Entry Annotation List $K : S \to \mathcal{P}(T)$
Path $\nu = ((t_1, r_1) \leq \cdots \leq (t_m, r_m))$ in P

Output:

Edge Entry Annotation List $L_\nu : \text{Edge}(\text{VR}(S, h)_\nu) \to \mathcal{P}(P)$

```
function GetPointOfEntry(List K(x), path ν)
    for every step (t_i, r_i) in the path ν do
        for every filtration step u in the point entry annotation K(x) do
            if u ≤ t_i then                          ▷ check if the point x is contained in S_{t_i}
                return (t_i, r_i)
            end if
        end for
    end for
    return ∞
end function

procedure GetEdgeEntryAnnotation
    Set L_ν to empty list
    for every pair of points x_i, x_j in S with i > j parallel do
        a := GetPointOfEntry(K(x_i), ν)
        b := GetPointOfEntry(K(x_j), ν)
        Append (max(a, b), H_ij) to L_ν
    end parallel do
end procedure
```

A parallelized implementation of the MuRiT algorithm, is available at https://github.com/tdalife/murit. This implementation combines Algorithms 1 and 2, and is optimized for the Vietoris-Rips persistence analysis of multi-filtered datasets.

4. An Application to Viral Evolution

4.1. Topology of Evolutionary Processes. A particularly useful application of the MuRiT algorithm is in the surveillance of pathogen evolution proposed in [BHPG+21]. While the evolution of an organism is usually modeled according to the paradigm of phylogenetic trees, there are various biological phenomena that are incompatible with this approach. In the example of viral evolution, an exchange of genomic information between distinct lineages can happen through recombination or reassortment, which is known to be a key factor in rapid host adaptation for many viruses [CCR13]. These instances of reticulate evolution can be viewed as a deviation from a trivial tree topology and obstruct the existence of a single phylogeny, as different parts of the genome might admit conflicting evolutionary histories.

In this context, we consider a finite set S of genome sequences, which we will think of as finite words $x = (x_1, x_2, \ldots, x_N)$ of uniform length N in the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{T}, \mathrm{G}\}$ corresponding to the four nucleotides, and endow it with a semimetric measuring the genetic distance between any pair of sequences. A standard choice of semimetric is given by the Hamming distance

$$h : S \times S \to \mathbb{N} \qquad (x, y) \mapsto h(x, y) := \#\{i \,|\, x_i \neq y_i\},$$

which counts the number of genomic positions at which two sequences differ. As put forward in [CCR13], persistent homology provides a fast and effective method to extract patterns of non-trivial topology from the genomic data set S. In this approach, evolutionary relationships at all scales are comprehensively encoded in the Vietoris-Rips filtration

$$\mathrm{VR}_0(S, h) \subseteq \mathrm{VR}_1(S, h) \subseteq \mathrm{VR}_2(S, h) \subseteq \ldots,$$

while information on both the tree-structure and reticulate events is captured in the persistence barcode.

4.2. Topological Recurrence Analysis. In addition to the exchange of genetic material, a further source of non-trivial topology in the phylogeny is convergent evolution. The corresponding reticulate events are called *homoplasies* and mean the independent acquisition of a specific mutation in different lineages. If the sampling of data is sufficiently dense, this typically gives rise to a persistent homology class at the smallest scale, admitting a cycle representative for which all edges correspond to *single nucleotide variations* (SNV) with edge length equal to 1. Recall at this point that a single nucleotide variation means a mutation of a single nucleotide at precisely one position in the genome. Such cycles admit a convenient description in terms of their location in the barcode. The representatives of persistent homology classes corresponding to bars in the barcode $\mathcal{B}_1(\mathrm{VR}(S, h))$ that are born in the first filtration step $\mathrm{VR}_1(S, h)$ will be called *SNV cycles*.

In [BHPG+21], SNV cycles are used to define a measure of convergent evolution based on one-parameter persistent homology. Let us briefly outline the main ideas of this approach. The barcode associated with the one-dimensional persistent homology $H_1(\mathrm{VR}(S, h))$ is computed

with a custom version of `Ripser` [Bau21b] that also has the capability to find cycle representatives. This method of *exhaustive reduction* (see [ZC05, EO19]) aims to produce cycles that tightly fit the data, by systematically replacing the longest edge of a given cycle with shorter edges of suitable nearby cycles. By definition of SNV cycles, for their extraction it suffices to consider persistent homology only on the smallest scale, which amounts to running `Ripser` with scale parameter threshold set to 2, leading to a substantial speedup of the computation. Thanks to the specific properties of the Hamming geometry of the semimetric space (S, h) in combination with the tree-like structure of the phylogeny, `Ripser` is able to process hundreds of thousands of data points in the particular case of SARS-CoV-2 evolution [BHPG$^+$21, BR22].

For each mutation in the viral genome, its *topological recurrence index* (tRI) is defined to be the total number of SNV cycles containing an edge that gives rise to the given mutation. Here we use the fact that each edge in an SNV cycle has length 1 and hence corresponds to a uniquely determined mutation in the genome. The topological recurrence index provides a lower bound for the number of independent occurrences of a mutation in the evolution of an organism and is therefore a measure for convergent evolution. As was demonstrated in [BHPG$^+$21], there is an abundance of SNV cycles in the SARS-CoV-2 genomic dataset and hence the above definition of the topological recurrence index, which only relies on SNV cycles, will already lead to statistical significant signals. It was moreover shown that the topological recurrence index can serve as an early indicator for emerging adaptive mutations in the evolution of the coronavirus.

In the case of ongoing genomic surveillance, the dataset admits a natural filtration by sampling time $S_1 \subseteq \cdots S_{m-1} \subseteq S_m$, where S_t denotes the set of all viral genomes sampled up until time step t. For every $t \in \{1, \ldots, m\}$, we denote by $\mathbf{SNV}_t$ the full set of SNV cycle representatives in time step t we get from the persistence analysis of (S_t, h) with `Ripser`. Note that, as a feature of Hamming geometry, the homology classes associated to SNV cycles are in a certain sense stable with respect to adding points in a time series: Inclusion of new data points might lead to a splitting of the cycle, but the resulting homology class is typically non-zero. It can happen that the homology class is destroyed by adding data points only when gaps or insertions in the genome alignment are involved in the SNV cycle.

In the study of SARS-CoV-2 evolution in [BHPG$^+$21], the barcode $\mathcal{B}_1(\mathrm{VR}(S_t, h))$ for each time step t is computed separately. Tracking SNV cycles over time yields information about the adaptation process of the pathogen. However, this can be troubled by the following two issues.

(1) Computing persistence homology at each time step separately can be time consuming and computationally expensive if the filtration by time of the dataset is large (like for example in a time series analysis over one year on a daily basis).

(2) SNV cycle representatives of the same homology class at different time steps will in general not be compatible with each other, which can lead to noise in the topological recurrence analysis. More precisely, if $\omega \in \mathbf{SNV}_t$ and if its image under the canonical morphism

$$H_1(\mathrm{VR}_1(S_t, h)) \to H_1(\mathrm{VR}_1(S_{t+1}, h))$$

induced by the inclusion $S_t \subseteq S_{t+1}$ is non-zero, then it may still happen that $\omega \notin \mathbf{SNV}_{t+1}$.

Both of these problems are resolved by the MuRiT algorithm in the following way. As explained in Section 3.4, we define the poset

$$P := \underbrace{\{1, \dots, m\}}_{\text{time steps}} \times \underbrace{h(S)}_{\text{distances}} .$$

Then the Vietoris-Rips complex $X = \mathrm{VR}(S, h)$ naturally becomes a P-filtered simplicial complex. As for the definition of the topological recurrence index it is only relevant to determine the time of birth of each SNV cycle, it suffices to compute the persistence barcode in homology degree one of the subcomplex $\mathrm{VR}(S, h)_\nu \subseteq \mathrm{VR}(S, h)$ determined by the path $\nu = ((1, 1) \leq \cdots \leq (m, 1))$ in P (see Figure 3). For this, we use `MuRiT` to compute the persistence barcode in homology degree one of the Vietoris-Rips transformation of $\mathrm{VR}(S, h)_\nu$. At each time step t, `Ripser` extracts a full set of SNV cycles such that the corresponding homology classes correspond to bars $I \in \mathcal{B}_1\big(\widehat{\mathrm{VR}}(\mathrm{VR}(S, h)_\nu)\big)$ with $t \in I$. In this way, we achieve compatibility of SNV cycles across all filtration steps. From a computational perspective, this means a great gain of efficiency as the persistence barcode of $H_1(\mathrm{VR}(S, h)_\nu)$, which resolves all time steps, is computed with a single run of `Ripser`.

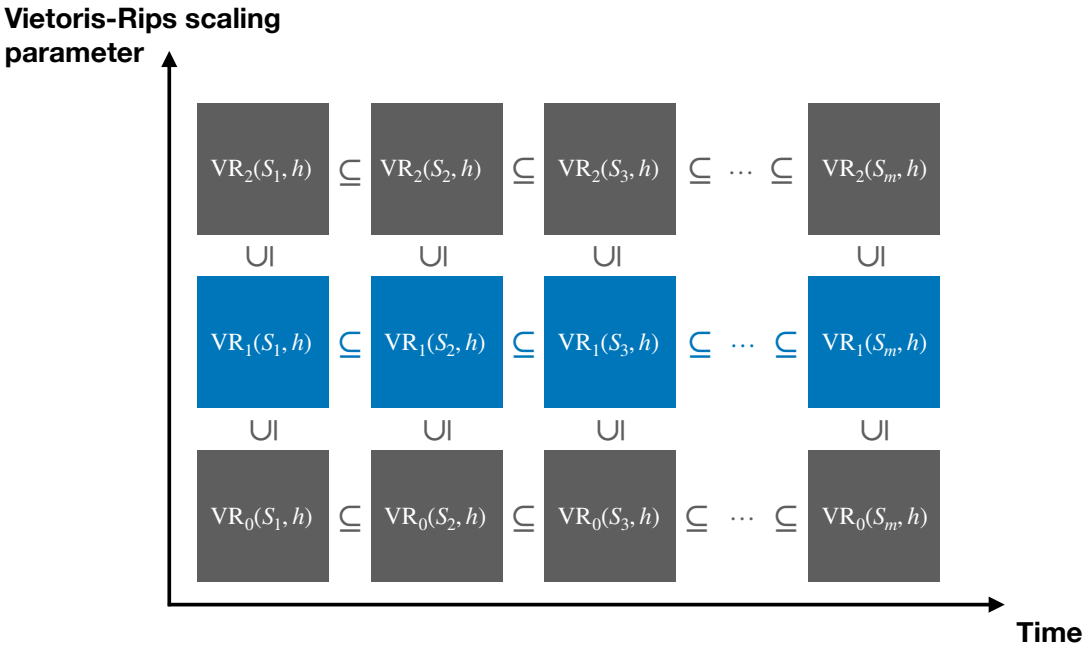

Figure 3: Pathwise subcomplexes and SNV cycles in viral evolution. The blue tiles mark the horizontal subcomplex $\mathrm{VR}(S, h)_\nu$ of the P-filtered Vietoris-Rips complex $\mathrm{VR}_\bullet(S_\bullet, h)$ corresponding to the path $\nu = ((1, 1) \leq \cdots \leq (m, 1))$ in $P = \{1, \dots, m\} \times h(S)$. This subcomplex keeps track of the formation of SNV cycles in the time-filtered dataset $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_m = S$ of viral gene sequences equipped with the Hamming distance h.

4.3. The CoVtRec Pipeline. The MuRiT algorithm is part of our CoVtRec pipeline for the early warning and surveillance of recurrent mutations in the evolution of the coronavirus SARS-CoV-2 in the current COVID-19 pandemic [BHNO22]. Regular reports containing analyses on the basis of SARS-CoV-2 genomic data shared via GISAID, the global data science initiative [SM17, KGF+21], are available at https://tdalife.github.io/covtrec. Recall that recurrent mutations are potentially adaptive in the sense that they could confer some fitness advantage to the virus, such as immune evasion or higher infectivity. In the current phase of the pandemic, the early identification of potentially adaptive mutations is of paramount importance as the virus is constantly developing new variants by mutating its genome. For more details on the biological aspects of the topological recurrence analysis of SARS-CoV-2 genomic data see [BHPG+21].

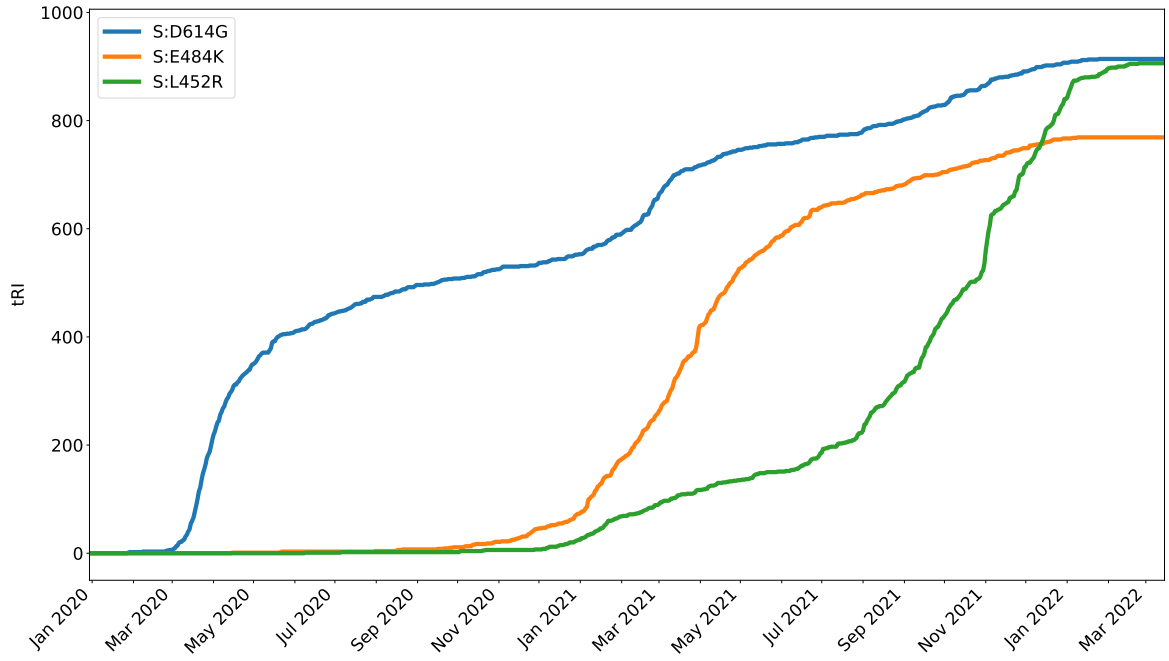

Figure 4. Surveillance of the convergent evolution of the coronavirus SARS-CoV-2. The diagram shows the time plots at daily resolution for the topological recurrence index (tRI) of the adaptive SARS-CoV-2 Spike gene mutations D614G, E484K and L452R over a period of 27 months, from the beginning of the pandemic in late December 2019 until 15 March 2022. The MuRiT algorithm enables the efficient topological analysis of hundreds of thousands of data points over time by leveraging the natural stratification by time of the coronavirus gene sequences dataset. While the mutation D614G is currently observed in essentially all virus samples, E484K occurred in the Alpha and Beta variants, and L452R has more recently been observed in the Delta and Omicron BA.5 variants.

The CoVtRec pipeline generates time series analysis charts for the topological recurrence index (tRI) at daily resolution by leveraging the natural stratification by time of SARS-CoV-2 genomic data. Thanks to highly optimized algorithms that take advantage of the tree-like structure of the data [BR22], CoVtRec can process very large SARS-CoV-2 genomic datasets and easily scales to hundreds of thousands of distinct genomes. This demonstrates the efficiency and usefulness of

the MuRiT algorithm in practice. To give a concrete example, we analyzed topological signals for the ongoing convergent evolution for three prominent mutations of the SARS-CoV-2 genome from the beginning of the pandemic in December 2019 until 15 March 2022. To that end, we performed a topological recurrence analysis for a curated alignment of 5,323,639 high-quality SARS-CoV-2 Spike gene sequences shared via GISAID. The analysis was restricted to the Spike gene, a part of the genome that determines the structure of the Spike protein on the surface of the virus and therefore plays an essential role in immune evasion and infectivity. Our algorithm performed the topological analysis of 359,650 distinct Spike gene sequences in less than a day on a machine with Intel Xeon Gold 6230R processors and 52 kernels.

We analyzed topological signals of convergence for the Spike gene mutations D614G, E484K and L452R (see Figure 4). All of these mutations exhibit a topological signal of convergence, with the topological recurrence index (tRI) rising over the course of the pandemic. We conclude that they are potentially adaptive. In fact, there is by now experimental evidence that the mutation D614G increases transmissibility [KFG+20, LWN+20] and in vitro infectiousness [PLL+21, HCH+20, YWP+20], and the mutations E484K and L452R enable the virus to evade immune protection [GLC+21, LVB+21]. While the mutation D614G superseded the wild type already at the beginning of the pandemic and is currently observed in essentially all virus samples, E484K occurred in the Alpha and Beta variants, and L452R has more recently been observed in the Delta and Omicron BA.5 variants [WHO]. A more detailed discussion of biological implications of the presence of topological signals for mutations in the evolution of the coronavirus is available in [BHPG+21].

Data Availability and Data Preparation. All SARS-CoV-2 genome data used in this work are available from the GISAID EpiCov Database [SM17, KGF+21] and are accessible online at https://doi.org/10.55876/gis8.220629ug. Our analysis of SARS-CoV-2 genome data is based on the alignment msa_0315.fasta downloaded from the GISAID EpiCoV Database [SM17, KGF+21] on 28 March 2022. This alignment comprises 8,297,154 SARS-CoV-2 whole genome sequences that have been aligned to the reference sequence Wuhan/WIV04 with GISAID accession number EPI_ISL_402124 using MAFFT (Version 7) [Kat02]. Sequences in this alignment were truncated to the Spike gene (reference site positions 21,563 to 25,384), and subsequently sequences containing any characters other than A, C, T, G or gaps or insertions represented by - were removed. This resulted in an alignment comprising 5,323,639 complete SARS-CoV-2 Spike genes of length 4,874nt.

References

[WHO] Tracking SARS-CoV-2 variants. Available at https://www.who.int/activities/tracking-SARS-CoV-2-variants.

[Bau21a] U. Bauer, Ripser: efficient computation of Vietoris–Rips persistence barcodes, *GitHub* (2015-2021), Comment on triangle inequality: https://github.com/Ripser/ripser/issues/12.

[Bau21b] U. Bauer, Ripser: efficient computation of Vietoris–Rips persistence barcodes, *Journal of Applied and Computational Topology* (2021). https://doi.org/10.1007/s41468-021-00071-5.

[BR22] U. Bauer and F. Roll, Gromov hyperbolicity, geodesic defect, and apparent pairs in vietoris-rips filtrations, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022 (en). https://doi.org/10.4230/LIPICS.SOCG.2022.15.

[BHNO22] M. Bleher, L. Hahn, M. Neumann, and A. Ott, TDALife - CoVtRec: Topological Surveillance of Recurrent Mutations in SARS-CoV-2, *GitHub* (2022), https://tdalife.github.io/covtrec/.

[BHPG+21] M. Bleher, L. Hahn, J. A. Patino-Galindo, M. Carriere, U. Bauer, R. Rabadan, and A. Ott, Topological data analysis identifies emerging adaptive mutations in SARS-CoV-2, 2021. https://doi.org/10.48550/ARXIV.2106.07292.

[BL22] M. B. Botnan and M. Lesnick, An introduction to multiparameter persistence, 2022. https://doi.org/10.48550/ARXIV.2203.14289.

[BOO21] M. B. Botnan, S. Oppermann, and S. Oudot, Signed barcodes for multi-parameter persistence via rank decompositions and rank-exact resolutions, 2021. https://doi.org/10.48550/ARXIV.2107.06800.

[Car09] G. Carlsson, Topology and data, *Bulletin of the American Mathematical Society* **46** no. 2 (2009), 255–308. https://doi.org/10.1090/s0273-0979-09-01249-x.

[CSZ09] G. Carlsson, G. Singh, and A. Zomorodian, Computing multidimensional persistence, in *Algorithms and Computation*, Springer Berlin Heidelberg, 2009, pp. 730–739. https://doi.org/10.1007/978-3-642-10631-6_74.

[CZ09] G. Carlsson and A. Zomorodian, The Theory of Multidimensional Persistence, *Discrete & Computational Geometry* **42** no. 1 (2009), 71–93. https://doi.org/10.1007/s00454-009-9176-0.

[CZCG05] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas, Persistence barcodes for shapes., *International Journal of Shape Modeling* **11** (2005), 149–188. https://doi.org/10.1145/1057432.1057449.

[CFF+13] A. Cerri, B. D. Fabio, M. Ferri, P. Frosini, and C. Landi, Betti numbers in multidimensional persistent homology are stable functions, *Mathematical Methods in the Applied Sciences* **36** no. 12 (2013), 1543–1557. https://doi.org/10.1002/mma.2704.

[CCR13] J. M. Chan, G. Carlsson, and R. Rabadan, Topology of viral evolution, *Proceedings of the National Academy of Sciences* **110** no. 46 (2013), 18566–18571. https://doi.org/10.1073/pnas.1313480110.

[CdSGO16] F. Chazal, V. de Silva, M. Glisse, and S. Oudot, *The structure and stability of persistence modules*, *SpringerBriefs in Mathematics*, Springer, [Cham], 2016. https://doi.org/10.1007/978-3-319-42545-0.

[DW22] T. K. Dey and Y. Wang, *Computational topology for data analysis*, Cambridge University Press, Cambridge, 2022. https://doi.org/10.1017/9781009099950.

[EH08] H. Edelsbrunner and J. Harer, Persistent homology—a survey, 2008, pp. 257–282. https://doi.org/10.1090/conm/453/08802.

[EH10] H. Edelsbrunner and J. L. Harer, *Computational topology*, American Mathematical Society, Providence, RI, 2010, An introduction. https://doi.org/10.1090/mbk/069.

[EO19] H. Edelsbrunner and K. Ölsböck, Holes and dependences in an ordered complex, *Computer Aided Geometric Design* **73** (2019), 1–15. https://doi.org/10.1016/j.cagd.2019.06.003.

[Ghr07] R. Ghrist, Barcodes: The persistent topology of data, *Bulletin of the American Mathematical Society* **45** no. 01 (2007), 61–76. https://doi.org/10.1090/s0273-0979-07-01191-3.

[GLC+21] A. J. Greaney, A. N. Loes, K. H. Crawford, T. N. Starr, K. D. Malone, H. Y. Chu, and J. D. Bloom, Comprehensive mapping of mutations in the SARS-CoV-2 receptor-binding domain that affect recognition by polyclonal human plasma antibodies, *Cell Host & Microbe* **29** no. 3 (2021), 463–476.e6. https://doi.org/10.1016/j.chom.2021.02.003.

[Han21] M. Hanussek, Valet, *GitHub* (2021), https://github.com/MaximilianHanussek/VALET.

[HCH+20] Y. J. Hou, S. Chiba, P. Halfmann, C. Ehre, M. Kuroda, K. H. Dinnon, and S. R. L. et al., SARS-CoV-2 d614g variant exhibits efficient replication ex vivo and transmission in vivo, *Science* **370** no. 6523 (2020), 1464–1468. https://doi.org/10.1126/science.abe8499.

[Kat02] K. Katoh, MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform, *Nucleic Acids Research* **30** no. 14 (2002), 3059–3066. https://doi.org/10.1093/nar/gkf436.

[KK21] L. Keegan and D. Kempf, Hammingdist: A Fast Tool to Calculate Hamming Distances, *GitHub* (2021), https://github.com/ssciwr/hammingdist.

[KGF+21] S. Khare, C. Gurry, L. Freitas, M. B. Schultz, G. Bach, A. Diallo, and N. A. et al., GISAID's role in pandemic response, *China CDC Weekly* **3** no. 49 (2021), 1049–1051. https://doi.org/10.46234/ccdcw2021.255.

[KFG+20] B. Korber, W. M. Fischer, S. Gnanakaran, H. Yoon, J. Theiler, W. Abfalterer, N. Hengartner, and E. E. G. et al., Tracking Changes in SARS-CoV-2 Spike: Evidence that D614G Increases Infectivity of the COVID-19 Virus, *Cell* **182** no. 4 (2020), 812–827.e19. https://doi.org/10.1016/j.cell.2020.06.043.

[LW15] M. Lesnick and M. Wright, Interactive Visualization of 2-D Persistence Modules, 2015. https://doi.org/10.48550/ARXIV.1512.00180.

[LWN+20] Q. Li, J. Wu, J. Nie, L. Zhang, H. Hao, S. Liu, and C. Z. et al., The impact of mutations in SARS-CoV-2 spike on viral infectivity and antigenicity, *Cell* **182** no. 5 (2020), 1284–1294.e9. https://doi.org/10.1016/j.cell.2020.07.012.

[LVB+21] Z. Liu, L. A. VanBlargan, L.-M. Bloyet, P. W. Rothlauf, R. E. Chen, S. Stumpf, H. Zhao, J. M. Errico, E. S. Theel, and M. J. e. a. Liebeskind, Identification of SARS-CoV-2 spike mutations that attenuate monoclonal and serum antibody neutralization, *Cell Host & Microbe* **29** no. 3 (2021), 477–488.e4. https://doi.org/10.1016/j.chom.2021.01.014.

[LGSL19] D. Luetgehetmann, D. Govc, J. Smith, and R. Levi, Computing persistent homology of directed flag complexes, *arXiv* (2019), DOI: 10.48550/ARXIV.1906.10458.

[MS05] E. Miller and B. Sturmfels, *Combinatorial Commutative Algebra*, 1 ed., Springer-Verlag New York, 2005. https://doi.org/10.1007/b138602.

[OPT+17] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, A roadmap for the computation of persistent homology, *EPJ Data Science* **6** no. 1 (2017). https://doi.org/10.1140/epjds/s13688-017-0109-5.

[Oud15] S. Y. Oudot, *Persistence theory: from quiver representations to data analysis*, *Mathematical Surveys and Monographs* **209**, American Mathematical Society, Providence, RI, 2015. https://doi.org/10.1090/surv/209.

[PLL+21] J. A. Plante, Y. Liu, J. Liu, H. Xia, B. A. Johnson, K. G. Lokugamage, and X. e. a. Zhang, Spike mutation D614G alters SARS-CoV-2 fitness, **592** no. 7852 (2021), 116–121. https://doi.org/10.1038/s41586-020-2895-3.

[SM17] Y. Shu and J. McCauley, GISAID: Global initiative on sharing all influenza data – from vision to reality, *Eurosurveillance* **22** no. 13 (2017). https://doi.org/10.2807/1560-7917.es.2017.22.13.30494.

[Wei11] S. Weinberger, What is. . .persistent homology?, *Notices Amer. Math. Soc.* **58** no. 1 (2011), 36–39.

[WLK+20] M. Wright, M. Lesnick, B. Keller, R. Zhao, S. Segert, D. Turner, A. Yu, A. De, P. Nadolny, and M. Abdel-Rahman, RIVET, Version 1.1, *GitHub* (2020), https://github.com/rivetTDA.

[YWP+20] L. Yurkovetskiy, X. Wang, K. E. Pascal, C. Tomkins-Tinch, T. P. Nyalile, Y. Wang, and A. B. et al., Structural and functional analysis of the d614g SARS-CoV-2 spike protein variant, *Cell* **183** no. 3 (2020), 739–751.e8. https://doi.org/10.1016/j.cell.2020.09.032.

[ZC05] A. Zomorodian and G. Carlsson, Computing Persistent Homology, *Discrete & Computational Geometry* **33** no. 2 (2005), 249–274. https://doi.org/10.1007/s00454-004-1146-y.