Lecture 15: Computing Betti Numbers

From boundary ranks to persistent H_1

Jordi-Lluís Figueras

Matematiska Institutionen, Uppsala Universitet

May 13, 2026

Where Lecture 14 stopped

At one fixed scale r, we have a simplicial complex K and chain groups

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0.$$

Cycles and boundaries are

$$Z_k = \ker \partial_k, \qquad B_k = \operatorname{im} \partial_{k+1}.$$

Homology is

$$H_k = Z_k / B_k.$$

The Betti number is the dimension of this quotient:

$$\beta_k = \dim H_k.$$

Why dimensions minus ranks appear

The identity

$$\partial_k \partial_{k+1} = 0$$

implies

$$B_k \subseteq Z_k.$$

Therefore $H_k = Z_k / B_k$ is obtained by declaring boundaries to be zero inside the vector space of cycles.

For finite-dimensional vector spaces,

$$\dim(Z_k / B_k) = \dim Z_k - \dim B_k.$$

So

$$\beta_k = \dim Z_k - \dim B_k.$$

The rank formula

By rank-nullity,

$$\dim Z_k = \dim \ker \partial_k = \dim C_k - \operatorname{rank}(\partial_k).$$

Also,

$$\dim B_k = \dim \operatorname{im} \partial_{k+1} = \operatorname{rank}(\partial_{k+1}).$$

Therefore

$$\boxed{\beta_k = \dim C_k - \operatorname{rank}(\partial_k) - \operatorname{rank}(\partial_{k+1}).}$$

This is the computational form of homology used today.

Lecture goals

In this lecture we will:

- repeat and justify the Betti-number rank formula;
- stress that β_0 counts connected components;
- compute β_1 from boundary matrices over $\mathbb{Z}_2$;
- compare graph cycles with homological holes;
- show computational examples for fixed-scale β_1;
- then explain how the same computation is organized across scales;
- read persistent H_1 intervals for a sensor example.

Fixed-Scale Betti Numbers

Count components and independent holes by ranks

The two formulas to remember

For $k = 0$, since $\partial_0 = 0$,

$$\boxed{\beta_0 = \dim C_0 - \operatorname{rank}(\partial_1).}$$

Interpretation of β_0

β_0 is the number of connected components.

For $k = 1$,

$$\boxed{\beta_1 = \dim C_1 - \operatorname{rank}(\partial_1) - \operatorname{rank}(\partial_2).}$$

This is the main computation today: number of independent edge cycles minus the cycles filled by triangles.

How to compute β_1 in practice

At one chosen scale r:

1. list all vertices, edges, and filled triangles of the complex;
2. order the edges and build $\partial_1 : C_1 \to C_0$;
3. order the triangles and build $\partial_2 : C_2 \to C_1$;
4. row-reduce both matrices over $\mathbb{Z}_2$;
5. substitute the ranks in

$$\beta_1 = \dim C_1 - \operatorname{rank}(\partial_1) - \operatorname{rank}(\partial_2).$$

Computational meaning

$\operatorname{rank}(\partial_1)$ removes edge chains that are not closed; $\operatorname{rank}(\partial_2)$ removes closed cycles that are already filled.

Example 1: a path has no H_1 hole

There are four vertices, three edges, and no triangles:

$$\dim C_0 = 4, \qquad \dim C_1 = 3, \qquad \dim C_2 = 0.$$

The graph is connected, so

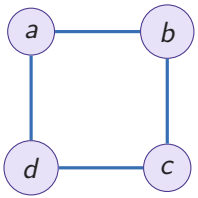
$$\operatorname{rank}(\partial_1) = 3.$$

There are no triangles, so $\operatorname{rank}(\partial_2) = 0$.
Thus

$$\beta_1 = 3 - 3 - 0 = 0.$$

A connected graph with no cycle has no one-dimensional hole.

Example 2: a square cycle



no filled triangles

There are four vertices and four edges:

$$\dim C_0 = 4, \qquad \dim C_1 = 4, \qquad \dim C_2 = 0.$$

The graph is connected, so

$$\operatorname{rank}(\partial_1) = 3.$$

There are no filled triangles, so

$$\operatorname{rank}(\partial_2) = 0.$$

Therefore

$$\beta_1 = 4 - 3 - 0 = 1.$$

Square cycle: the matrix behind the count

Order the vertices and edges as

$$a, b, c, d, \qquad ab, bc, cd, da.$$

Then

$$\partial_1 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Gaussian elimination over $\mathbb{Z}_2$ gives

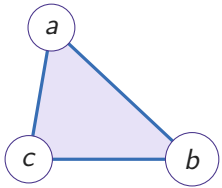
$$\operatorname{rank}(\partial_1) = 3.$$

Hence

$$\dim \ker \partial_1 = \dim C_1 - \operatorname{rank}(\partial_1) = 4 - 3 = 1.$$

This one-dimensional kernel is generated by the square cycle.

Example 3: a filled triangle



There are three vertices, three edges, and one filled triangle:

$$\dim C_0 = 3, \qquad \dim C_1 = 3, \qquad \dim C_2 = 1.$$

The graph is connected, so

$$\operatorname{rank}(\partial_1) = 2.$$

The filled triangle has nonzero boundary, so

$$\operatorname{rank}(\partial_2) = 1.$$

Therefore

$$\beta_1 = 3 - 2 - 1 = 0.$$

Filled triangle: the cycle is killed by ∂_2

Order the edges as

$$ab, ac, bc.$$

The boundary of the filled triangle is

$$\partial_2([a, b, c]) = ab + ac + bc.$$

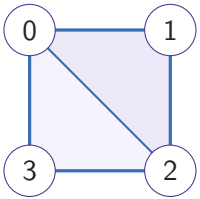
As a matrix,

$$\partial_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \operatorname{rank}(\partial_2) = 1.$$

The edge cycle still lies in $\ker \partial_1$, but it also lies in $\operatorname{im} \partial_2$.

Thus it is a boundary, not a homological hole.

Example 4: a square filled by two triangles



Simplices:

- four vertices;
- five edges: $01, 12, 23, 30, 02$;
- two filled triangles: $012, 023$.

The graph is connected, so

$$\mathrm{rank}(\partial_1) = 3.$$

The two triangle boundaries are independent, so

$$\mathrm{rank}(\partial_2) = 2.$$

Therefore

$$\beta_1 = 5 - 3 - 2 = 0.$$

Why the diagonal does not create a new hole

With edge order

$$01, 12, 23, 30, 02,$$

the two columns of ∂_2 are

$$\partial_2(012) = 01 + 12 + 02, \qquad \partial_2(023) = 23 + 30 + 02.$$

Thus

$$\partial_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}, \qquad \mathrm{rank}(\partial_2) = 2.$$

The square boundary equals the sum of the two triangle boundaries:

$$(01 + 12 + 02) + (23 + 30 + 02) = 01 + 12 + 23 + 30.$$

The diagonal appears twice and cancels over $\mathbb{Z}_2$.

Graph-only shortcut and its limitation

If the complex has no filled triangles, then $C_2 = 0$ and

$$\beta_1 = \dim C_1 - \operatorname{rank}(\partial_1).$$

For a graph with n vertices, m edges, and c connected components,

$$\operatorname{rank}(\partial_1) = n - c.$$

Hence for a pure graph,

$$\beta_1 = m - n + c.$$

But Vietoris–Rips complexes usually contain filled triangles. Then the correct formula is

$$\beta_1 = m - \operatorname{rank}(\partial_1) - \operatorname{rank}(\partial_2).$$

The $\operatorname{rank}(\partial_2)$ term is exactly what removes cycles filled by triangles.

Computational message before persistence

At a fixed scale, computing β_1 is not a visual task.

It is a linear-algebra task over $\mathbb{Z}_2$:

$$\text{build } \partial_1, \partial_2 \quad \longrightarrow \quad \text{row-reduce} \quad \longrightarrow \quad \beta_1.$$

Main point

β_1 counts independent closed edge cycles that are not boundaries of filled 2-chains.

The persistent version repeats this idea while the scale parameter changes.

Persistent Homology

Which holes survive across scales?

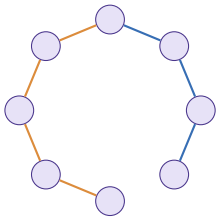
Why one scale is not enough

In a metric data set, the graph depends on a threshold r:

$$d(x_i, x_j) \leq r \quad \Longrightarrow \quad \text{edge } ij.$$

If r is too small, the graph may be disconnected.

If r is too large, almost everything becomes filled.



the visible topology depends on scale

So the question is not only:

What is H_1 at this scale?

but also:

Which H_1 features persist as r changes?

Filtration

For a point cloud X, increasing the scale produces nested complexes:

$$r_1 \leq r_2 \quad \Longrightarrow \quad \mathrm{VR}(X, r_1) \subseteq \mathrm{VR}(X, r_2).$$

A nested sequence

$$K_1 \subseteq K_2 \subseteq K_3 \subseteq \cdots$$

is called a filtration.

For Vietoris–Rips complexes, the filtration comes from the scale parameter r.

As r increases, simplices are added but never removed.

What changes as the scale grows?

As r increases:

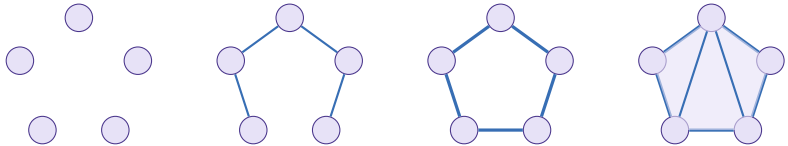
1. vertices begin isolated;
2. edges appear between nearby points;
3. connected components merge;
4. loops appear;
5. triangles fill some loops;
6. higher-dimensional filled structures appear.

For H_1, persistent homology tracks:

Birth and death

When a loop appears, and when it becomes filled.

A growing Vietoris–Rips complex



isolated points $\rightarrow$ partial graph $\rightarrow$ loop born $\rightarrow$ loop filled

Birth of a class

An H_1 class is born at the first scale where a new independent loop appears.

At that scale, there is a cycle

$$z \in Z_1$$

that is not yet a boundary:

$$z \notin B_1.$$

Geometrically:

> the edges have closed a loop, but the interior has not yet been filled.

The birth scale records when this feature first becomes visible.

Death of a class

An H_1 class dies when the corresponding loop becomes a boundary of filled 2-simplices.

At the death scale, new triangles have appeared so that

$$z = \partial A$$

for some 2-chain A.

Geometrically:

the loop has been filled by triangles.

The persistence or lifetime is

$$\text{death} - \text{birth}.$$

Barcode

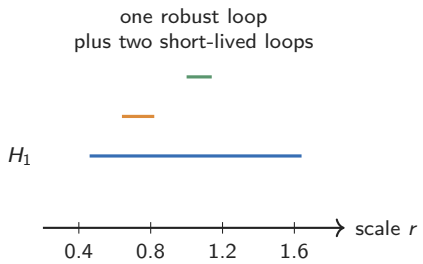
A barcode represents each persistent homology class by an interval:

$$[\text{birth}, \text{death}).$$

For H_1:

- the left endpoint is where a loop appears;
- the right endpoint is where it is filled;
- long intervals are more robust across scales.

Short bars often represent small geometric accidents or noise.



Persistence as robustness

Persistent homology changes the question.

Instead of asking:

Is there a hole at exactly this threshold?

we ask:

Is there a hole that survives over a meaningful range of thresholds?

Main interpretation

A persistent H_1 class is evidence of a robust loop in the pairwise-relation structure.

This is the topological analogue of distinguishing a stable pattern from a single-scale artifact.

Careful sensor interpretation

In the sensor-network example, a nonzero H_1 class in a Vietoris–Rips complex does not automatically prove a physical uncovered region.

The safer statement is:

Safe interpretation

Persistent H_1 detects a robust loop in the connectivity structure.

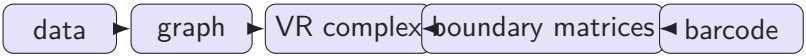
For precise coverage guarantees, one usually needs Cech complexes or comparison theorems between Cech and Vietoris–Rips complexes.

For this course, Vietoris–Rips complexes are useful because they are built directly from pairwise distances or pairwise connectivity.

Computation

From points to boundary matrices to barcodes

Computational pipeline



The script for this lecture is:

`scripts/vrPersistentHomology.py`

It is designed to show both sides of the method:

- a library computation using GUDHI;
- a small manual boundary-matrix computation over $\mathbb{Z}_2$.

Minimal GUDHI computation

The computational core is short:

```
rips = gd.RipsComplex(points=points, max_edge_length=1.8)
simplex_tree = rips.create_simplex_tree(max_dimension=2)
simplex_tree.compute_persistence(homology_coeff_field=2)
```

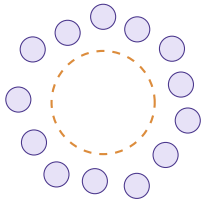
The last line computes persistent homology over $\mathbb{Z}_2$.

Then one can inspect:

```
simplex_tree.betti_numbers()
simplex_tree.persistence_intervals_in_dimension(1)
```

The library does large-scale boundary-matrix reduction; the mathematics is the same as in Lecture 14.

Synthetic data: sensor ring



sensors around a central region

The script generates a simplified sensor deployment:

$$x_i \approx (\cos\theta_i, \sin\theta_i).$$

The central dashed circle is only a visual guide. The computation uses the sensor positions and pairwise distances.

It builds the Vietoris–Rips filtration and computes persistent H_1.

Expected qualitative result:

- small scale: disconnected points;
- intermediate scale: one loop;
- large scale: triangles fill the loop.

Fixed-scale graph and homology

At a chosen scale r, the script plots edges satisfying

$$\|x_i - x_j\| \leq r.$$

It also constructs all triangles whose three edges are present and forms the boundary matrices

$$\partial_1 : C_1 \to C_0, \qquad \partial_2 : C_2 \to C_1.$$

Then it prints

$$\beta_0 = \dim C_0 - \operatorname{rank}(\partial_1), \qquad \beta_1 = \dim C_1 - \operatorname{rank}(\partial_1) - \operatorname{rank}(\partial_2).$$

Important distinction

The barcode is computed from the whole filtered complex, including filled triangles, not only from the plotted graph.

Manual rank computation over $\mathbb{Z}_2$

The script includes a small function `rankMod2(A)`.

It performs Gaussian elimination modulo 2, using only row swaps and row additions.

It then checks the fixed-scale formula

$$\beta_1 = \dim C_1 - \operatorname{rank}(\partial_1) - \operatorname{rank}(\partial_2)$$

on the examples from the beginning of the lecture.

Example	$\dim C_1$	ranks used	β_1
Path graph	3	$\operatorname{rank} \partial_1 = 3$	0
Square cycle	4	$\operatorname{rank} \partial_1 = 3$	1
Filled triangle	3	$\operatorname{rank} \partial_1 = 2,\ \operatorname{rank} \partial_2 = 1$	0
Filled square	5	$\operatorname{rank} \partial_1 = 3,\ \operatorname{rank} \partial_2 = 2$	0

Persistent homology applies the same linear-algebra idea to a filtered boundary matrix.

Running the script

From the lecture directory:

```
python3 scripts/vrPersistentHomology.py
```

For a non-interactive run that saves figures:

```
python3 scripts/vrPersistentHomology.py --no-show --save-dir figures
```

Required packages:

```
numpy matplotlib gudhi
```

If GUDHI is not installed, the script still computes the selected-scale boundary matrices and explains what persistence step is missing.

Sensor Problem

Detecting robust uncovered regions

The sensor problem across scales

We deploy sensors in a region and know their positions, or at least their pairwise detection distances.

For each scale r, we form the graph

$$d(x_i, x_j) \leq r \quad \Longleftrightarrow \quad \{i, j\} \in E_r.$$

Then we fill all cliques to obtain

$$\mathrm{VR}(X, r).$$

The practical question is not whether one hand-picked r gives a loop, but whether a loop is visible over a stable interval of scales.

Coverage, Čech, and Vietoris–Rips

There are two related but different objects.

Object	Built from	Meaning
Coverage disks	sensor locations and radius	physical covered region
Čech complex	common overlaps of disks	nerve of the cover
Vietoris–Rips complex	pairwise distances	computable clique complex

The Čech complex is the natural object for exact coverage statements.

The Vietoris–Rips complex is easier to compute from pairwise data and is the one used in our script.

Persistent H_1 for sensors

A persistent H_1 class in the sensor complex means:

Computational interpretation

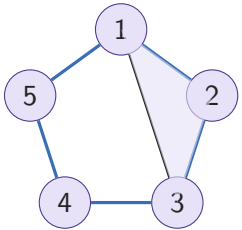
There is a loop in the pairwise detection structure that is not immediately filled by local triangles.

In a geometric deployment this may indicate sensors arranged around an uncovered central region.

But the rigorous conclusion depends on the model:

- for exact coverage, use a Čech or alpha-complex formulation;
- for pairwise network data, persistent Vietoris–Rips H_1 is a robust connectivity diagnostic.

Mini example: five sensors



one triangle fills part of the cycle

Suppose five sensors form a cycle of pairwise detections.

If every adjacent pair detects each other but only one triple is mutually connected, the clique complex may still contain an H_1 feature.

As the threshold changes, persistence asks whether this cyclic structure remains visible or quickly disappears.

What the script demonstrates

The script performs three computations on the same sensor deployment.

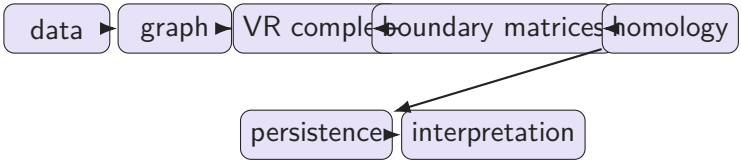
1. Build a fixed-scale Vietoris–Rips complex.
2. Compute β_0 and β_1 from boundary-matrix ranks over $\mathbb{Z}_2$.
3. Use GUDHI to compute persistent H_1 intervals across all scales up to a maximum edge length.

The expected output is one dominant H_1 interval corresponding to the central sensor loop.

Shorter intervals, if present, should be interpreted as scale-dependent local artifacts.

Summary Pipeline

Full pipeline



Final slogan

Vietoris–Rips homology turns local pairwise relations into computable information about global structure.

What to remember

- β_0 is the number of connected components.
- $\beta_1 = \dim C_1 - \operatorname{rank}(\partial_1) - \operatorname{rank}(\partial_2)$ is the main fixed-scale computation.
- The term $\operatorname{rank}(\partial_2)$ removes edge cycles filled by triangles.
- A filtration is a nested family of complexes.
- Vietoris–Rips filtrations arise by increasing a distance threshold.
- Persistent homology records births and deaths of homology classes.
- Long barcode intervals represent features that survive across scales.
- For H_1, persistence tracks loops that appear and later become filled.
- GUDHI computes persistence by reducing boundary matrices over finite fields.
- In the sensor problem, persistent H_1 is evidence of a robust loop in the detection structure.

Exercises

1. For the path, square, filled triangle, and filled square examples, reproduce the β_1 computation by hand.
2. Run the script and compare the printed manual β_1 checks with the lecture slides.
3. Run the script on the default sensor ring and identify the longest H_1 interval.
4. Change the selected graph scale. When does the visible loop appear, and when is it filled?
5. Move one sensor toward the center. Does the longest H_1 interval become shorter?

Suggested bibliography

For students who want to read more:

- H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*, American Mathematical Society, 2010.
- R. Ghrist, "Barcodes: The persistent topology of data," *Bulletin of the AMS*, 45, 61–75, 2008.
- V. de Silva and R. Ghrist, "Coordinate-free coverage in sensor networks," *Algebraic & Geometric Topology*, 7, 339–358, 2007.
- GUDHI documentation: `https://gudhi.inria.fr/`.

Acknowledgment

These slides were prepared with the assistance of OpenAI Codex / ChatGPT 5.5 as a drafting and editing tool.
All mathematical, pedagogical, and course-specific content remains under the responsibility of the lecturer.