Arnau Mas

September 3rd 2020

# What is radiomics?

- Has its origins in the field of oncology: tumour imaging

# What is radiomics?

- Has its origins in the field of oncology: tumour imaging
- Extract many radiomic features from scans

# What is radiomics?

- Has its origins in the field of oncology: tumour imaging
- Extract many radiomic features from scans
- Look for correlations between radiomic features and treatment response

# What is radiomics?

## Benefits

- Non-invasive
- More systematic
- Quantitative

# What is radiomics?

### Benefits

- Non-invasive
- More systematic
- Quantitative

### Drawbacks

- May be hard to reproduce
- High dimensional feature spaces
- Limited number of cases

# TOPiomics

- TOPiomics is an ongoing collaboration between the Computer Vision Center (CVC) and the Vall d'Hebron Institute of Oncology (VHIO).

# TOPiomics

- TOPiomics is an ongoing collaboration between the Computer Vision Center (CVC) and the Vall d'Hebron Institute of Oncology (VHIO).
- Main aim: early detection of patients with strange radiomic signatures, *outliers*, for whom standard treatments could fail.

# TOPiomics

- TOPiomics is an ongoing collaboration between the Computer Vision Center (CVC) and the Vall d'Hebron Institute of Oncology (VHIO).
- Main aim: early detection of patients with strange radiomic signatures, *outliers*, for whom standard treatments could fail.
- Use *topological data analysis*, hence TOPiomics (topological radiomics).

# Prior work

- Progress thus far: use the *mutual $k$-nearest neighbours* (M$k$NN ) graph to detect outliers.

# Prior work

- Progress thus far: use the *mutual $k$-nearest neighbours* (M$k$NN ) graph to detect outliers.
- Problem: there is a parameter that has to be fixed. It has to do with the scale at which we look at the data.

# Prior work

- Progress thus far: use the *mutual $k$-nearest neighbours* (M$k$NN ) graph to detect outliers.
- Problem: there is a parameter that has to be fixed. It has to do with the scale at which we look at the data.
- Idea: look at *every* parameter value.

# Prior work

- Progress thus far: use the *mutual $k$-nearest neighbours* (M$k$NN ) graph to detect outliers.
- Problem: there is a parameter that has to be fixed. It has to do with the scale at which we look at the data.
- Idea: look at *every* parameter value.

This is persistence

# The elevator pitch

Algebraic topology studies topological spaces by computing algebraic invariants, namely with functors

$$\mathsf{Top} \to \mathsf{Grp}$$
$$\mathsf{Top} \to \mathsf{Ab}$$
$$\mathsf{Top} \to \mathsf{Ring}$$
$$\mathsf{Top} \to \mathsf{Vect}_K$$
$$\vdots$$

# The elevator pitch

Algebraic topology studies topological spaces by computing algebraic invariants, namely with functors

$$\mathsf{Top} \to \mathsf{Grp}$$
$$\mathsf{Top} \to \mathsf{Ab}$$
$$\mathsf{Top} \to \mathsf{Ring}$$
$$\mathsf{Top} \to \mathsf{Vect}_K$$
$$\vdots$$

Homology groups are one of these functors,

$$H_n \colon \mathsf{Top} \to \mathsf{Ab}.$$

# The elevator pitch

Algebraic topology studies topological spaces by computing algebraic invariants, namely with functors

$$\mathsf{Top} \to \mathsf{Grp}$$
$$\mathsf{Top} \to \mathsf{Ab}$$
$$\mathsf{Top} \to \mathsf{Ring}$$
$$\mathsf{Top} \to \mathsf{Vect}_K$$
$$\vdots$$

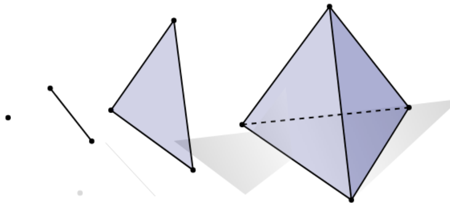
Homology groups are one of these functors,

$$H_n \colon \mathsf{Top} \to \mathsf{Ab}.$$

They come in many flavours, the simplest one is *simplicial homology*, which works for *simplicial complexes*.

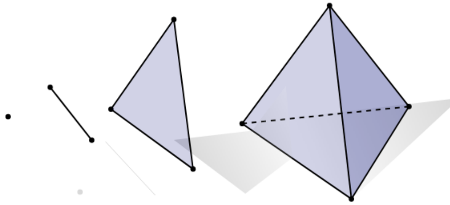
# The ingredients of homology

- *Simplex*. The convex hull of a set of points.



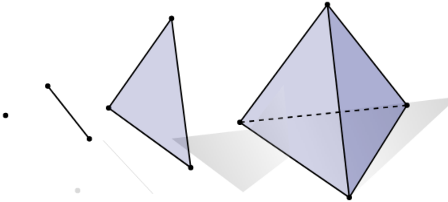
# The ingredients of homology

- *Simplex*. The convex hull of a set of (geometrically independent) points.



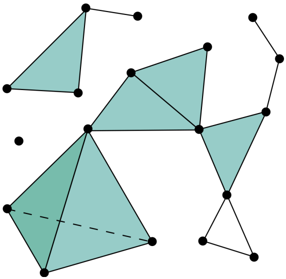
# The ingredients of homology

- *Simplex*. The convex hull of an (ordered) set of (geometrically independent) points.



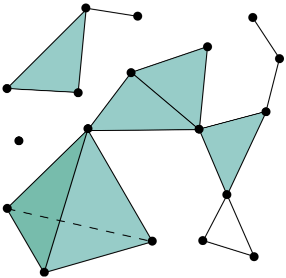
# The ingredients of homology

- *Simplex*. The convex hull of an (ordered) set of (geometrically independent) points.
- *Simplicial complex*, $K$. Topological space assembled out of simplices glued along their faces.



# The ingredients of homology

- *Simplex*. The convex hull of an (ordered) set of (geometrically independent) points.
- *Simplicial complex*, $K$. Topological space assembled out of simplices glued along their faces.
- *Chain groups*, $C_n(K)$. Free abelian groups generated by the $n$-simplices of a complex.



## Remark

In the combinatorial picture, the geometrical requirements are dropped and one speaks of *abstract* simplices, *abstract* simplicial complexes, etc.

# The boundary morphism

### Orientation

The simplex generated by $p_0, \ldots, p_n$ is written $[p_0, \ldots, p_n]$. The ordering determines an orientation, and we require that flipping orientation implies a change of sign in $C_n(K)$, e.g.

$$[p_0, p_1, \ldots, p_n] = -[p_1, p_0, \ldots, p_n].$$

# The boundary morphism

## Orientation

The simplex generated by $p_0, \ldots, p_n$ is written $[p_0, \ldots, p_n]$. The ordering determines an orientation, and we require that flipping orientation implies a change of sign in $C_n(K)$, e.g.

$$[p_0, p_1, \ldots, p_n] = -[p_1, p_0, \ldots, p_n].$$

The chain groups are not really free!

# The boundary morphisms

Define the boundary morphism $\partial_n \colon C_n(K) \to C_{n-1}(K)$ by

$$\partial_n[p_0, \ldots, p_n] := \sum_{k=0}^{n} (-1)^k [p_0, \ldots, \hat{p}_k, \ldots, p_n]$$

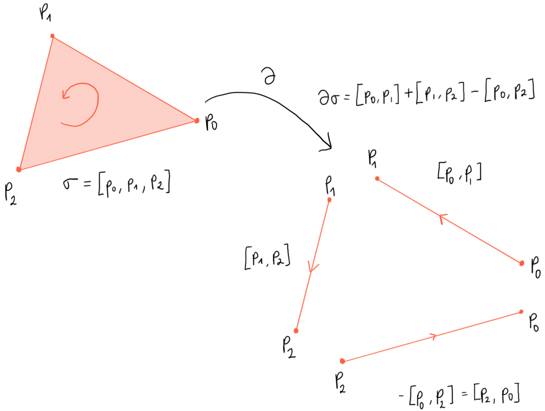
and extending to chains.

# The boundary morphisms

Define the boundary morphism $\partial_n \colon C_n(K) \to C_{n-1}(K)$ by

$$\partial_n[p_0, \dots, p_n] := \sum_{k=0}^{n} (-1)^k [p_0, \dots, \hat{p}_k, \dots, p_n]$$

and extending to chains.



# Boundaries and cycles

- *Cycles:* chains with no boundary,

$$Z_n(K) := \ker \partial_n \subseteq C_n(K)$$

# Boundaries and cycles

- *Cycles:* chains with no boundary,

$$Z_n(K) := \ker \partial_n \subseteq C_n(K)$$

- *Boundaries:* chains which are the boundary of a higher-dimensional chain,

$$B_n(K) := \operatorname{im} \partial_{n+1} \subseteq C_n(K)$$

# Boundaries and cycles

- *Cycles:* chains with no boundary,

$$Z_n(K) := \ker \partial_n \subseteq C_n(K)$$

- *Boundaries:* chains which are the boundary of a higher-dimensional chain,

$$B_n(K) := \operatorname{im} \partial_{n+1} \subseteq C_n(K)$$

## Fact

$$\partial_n \circ \partial_{n+1} = 0$$

so boundaries are always cycles.

# Homology groups

Homology groups are the quotients

$$H_n(K) := Z_n(K)/B_n(K).$$

# Homology groups

Homology groups are the quotients

$$H_n(K) := Z_n(K)/B_n(K).$$

The key insight: Homology groups measure cycles which are not boundaries.

# Homology groups

Homology groups are the quotients

$$H_n(K) := Z_n(K)/B_n(K).$$

The key insight: Homology groups measure cycles which are not boundaries. They tell us about the number of $n$-dimensional holes in our space.

# What is persistence?

Instead of a single simplicial complex, consider a *filtration*,

$$K^0 \subseteq K^1 \subseteq \cdots \subseteq K^N.$$

# What is persistence?

Instead of a single simplicial complex, consider a *filtration*,

$$K^0 \subseteq K^1 \subseteq \cdots \subseteq K^N.$$

Then look at this evolution at the level of homology

$$H^0_d(K) \hookrightarrow H^1_d(K) \hookrightarrow \cdots \hookrightarrow H^N_d(K)$$

# What is persistence?

Instead of a single simplicial complex, consider a *filtration*,

$$K^0 \subseteq K^1 \subseteq \cdots \subseteq K^N.$$

Then look at this evolution at the level of homology

$$H_d^{0}(K) \hookrightarrow H_d^{1}(K) \hookrightarrow \cdots \hookrightarrow H_d^{N}(K)$$

The steps of the filtration can be thought of as time

# What is persistence?

Because of functoriality, there are maps between the different steps of homology,

$$f_i^j\colon H_d^i(K) \to H_d^j(K).$$

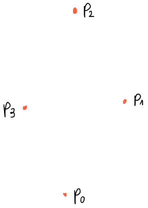
# What is persistence?

Because of functoriality, there are maps between the different steps of homology,

$$f_i^j : H_d^i(K) \to H_d^j(K).$$

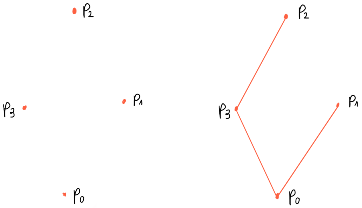
At each step, classes can be born (not in the image of $f_i^{i+1}$), die (in the kernel of $f_i^{i+1}$) or merge (have the same image through $f_i^{i+1}$).

# An example



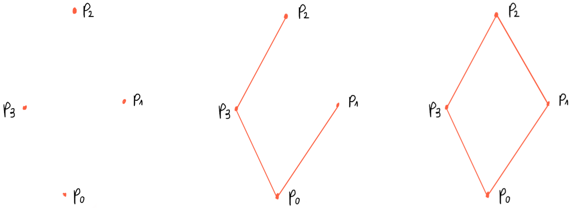

$[p_0], [p_1], [p_2], [p_3]$ are born.

# An example



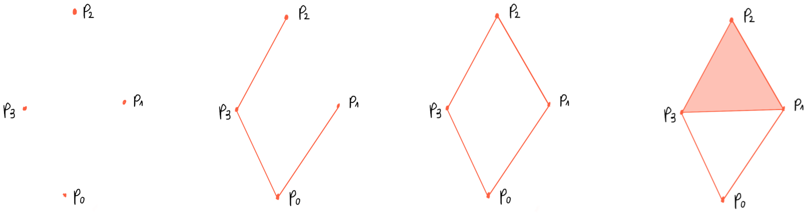

$[p_0], [p_1], [p_2], [p_3]$ merge.

# An example




The cycle $[p_0, p_1] + [p_1, p_2] + [p_2, p_3] + [p_3, p_4]$ is born.

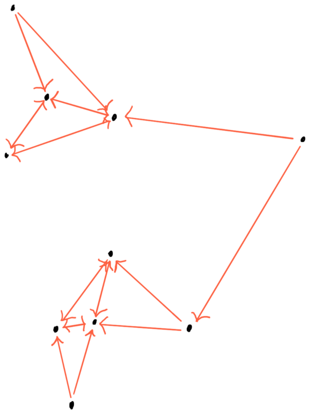
# An example

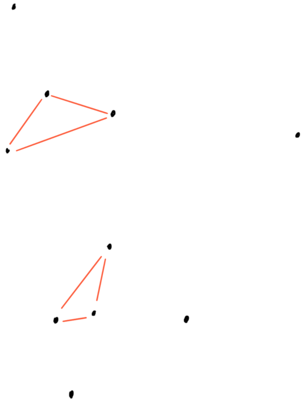



The cycle does not die.

# Why the M$k$NN graph?

It is good for detecting clusters in a data set.





# The outlier detection algorithm

The algorithm has four main parts:

# The outlier detection algorithm

The algorithm has four main parts:

- The class `Cloud` stores a point cloud and computes its M$k$NN graph.

# The outlier detection algorithm

The algorithm has four main parts:

- The class `Cloud` stores a point cloud and computes its M$k$NN graph.
- The class `Filtration` builds up the filtration.

# The outlier detection algorithm

The algorithm has four main parts:

- The class `Cloud` stores a point cloud and computes its M$k$NN graph.
- The class `Filtration` builds up the filtration.
- The class `Homology` computes the homology.

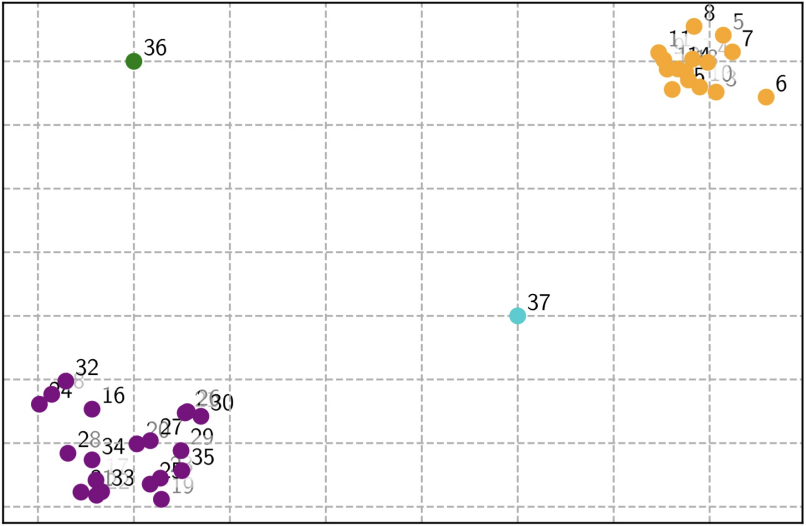
# The outlier detection algorithm

The algorithm has four main parts:

- The class `Cloud` stores a point cloud and computes its M$k$NN graph.
- The class `Filtration` builds up the filtration.
- The class `Homology` computes the homology.
- The persistence data of each class is gathered and plotted to detect the outliers.

# The class `Cloud`

# A run with toy data



# A run with toy data