Τοπολογική Ανάλυση Δεδομένων

Εαρινό Εξάμηνο 2026

Φροντιστηριακή Διάλεξη 2: Απλοτοπικά Συμπλέγματα Από Πίνακες Ζευγαρωτών Αποστάσεων, Κυβικά Συμπλέγματα & Εισαγωγή Στην Επίμονη Ομολογία

Διδάσκοντες:

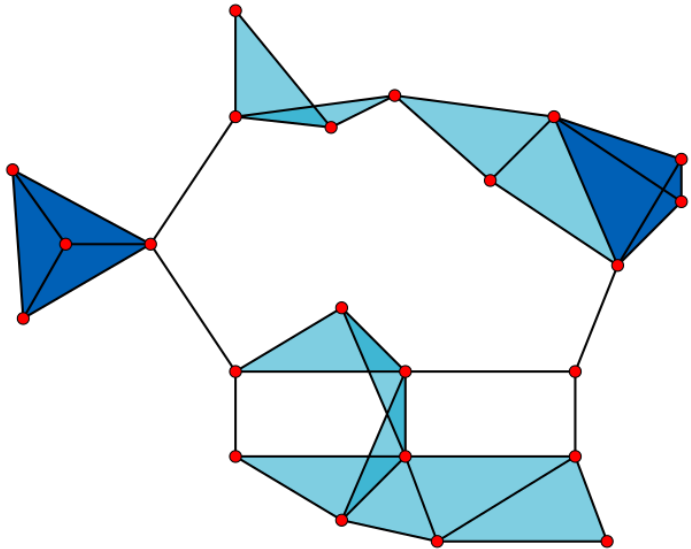
Αθανάσιος Ανδρικόπουλος

Ιωάννης Γουναρίδης

Επικουρικό Έργο: Αλέξανδρος - Μάριος Αφράτης

Πάτρα, 12 Μαρτίου 2026

Πίνακας Περιεχομένων



Απλοτοπικά Συμπλέγματα Από Πίνακες Ζευγαρωτών Αποστάσεων

Στόχος (Goal)

Κατασκευή Vietoris–Rips και Alpha συμπλεγμάτων από **πίνακα ζευγαρωτών αποστάσεων** (pairwise distance matrix), χωρίς άμεση πρόσβαση σε συντεταγμένες Ευκλείδειου χώρου.

Βιβλιοθήκες που Χρησιμοποιούνται

- `numpy` – αριθμητικοί υπολογισμοί
- `pandas` – ανάγνωση & επεξεργασία δεδομένων
- `gudhi` – κατασκευή συμπλεγμάτων
- `sklearn` – Πολυδιάστατη Κλιμάκωση (MDS)
- `matplotlib` – οπτικοποίηση

Εισαγωγή Βιβλιοθηκών (Importing Libraries)

Εισαγωγές για την Ενότητα 1

```
1       import numpy as np
2       import pandas as pd
3       import pickle as pickle
4       import gudhi as gd
5       from pylab import *
6
```

Σημείωση για `pylab`

Η `pylab` συνδυάζει `matplotlib.pyplot` και `numpy` σε ένα ενιαίο namespace – βολική για διαδραστική εξερεύνηση δεδομένων.

Σύνολο Δεδομένων: Σύνδεση Πρωτεΐνης (Protein Binding Dataset)

Περιγραφή (Description)

Μελετάμε διαμορφώσεις σύνδεσης της πρωτεΐνης δέσμευσης μαλτόζης (**MBP** – Maltose Binding Protein), ενός βιομορίου με 370 αμινοξικά κατάλοιπα (amino-acid residues).

Μετρική Δυναμικών Αποστάσεων (Dynamical Distances)

Η ανάλυση βασίζεται σε **δυναμικές αποστάσεις** (dynamical distances):

$$D_{ij} = 1 - |C_{ij}|$$

όπου C είναι ο **πίνακας συσχέτισης** (correlation matrix) μεταξύ καταλοίπων.

Σχήμα (Form) της Πρωτεΐνης

- **Κλειστή μορφή** (closed form): 7 διαμορφώσεις
- **Ανοιχτή μορφή** (open form): 7 διαμορφώσεις
- Σύνολο: **14 πίνακες συσχέτισης**

Βιβλιογραφία

Kovacev-Nikolic et al., *Stat Appl Genet Mol Biol*, 2016.

Φόρτωση & Προεπεξεργασία Δεδομένων (Data Loading & Preprocessing)

Ανάγνωση 14 Πινάκων Συσχέτισης

```
path_file = "./datasets/Corr_ProteinBinding/"
files_list = [
'1anf.corr_1.txt', '1ez9.corr_1.txt', '1fqa.corr_2.txt',
'1fqb.corr_3.txt', '1fqc.corr_2.txt', '1fqd.corr_3.txt',
'1jw4.corr_4.txt', '1jw5.corr_5.txt', '1lls.corr_6.txt',
'1mpd.corr_4.txt', '1omp.corr_7.txt', '3hpi.corr_5.txt',
'3mbp.corr_6.txt', '4mbp.corr_7.txt'
]
corr_list = [
pd.read_csv(path_file + u, header=None, sep='\s+')
for u in files_list
]

```

Υπολογισμός Πινάκων Αποστάσεων (Distance Matrices)

```
# Εφαρμογή του τύπου: D_ij = 1 - |C_ij|
dist_list = [1 - np.abs(c) for c in corr_list]
D = dist_list[0]    # Πρώτος πίνακας αποστάσεων (370 x 370)
D.head()            # Εκτύπωση πρώτων γραμμών

```

Φίλτρο Vietoris–Rips από Πίνακα Αποστάσεων (Rips Filtration from Distance Matrix)

Ορισμός

*Το **φίλτρο Vietoris–Rips** (Vietoris–Rips filtration) ενός μετρικού χώρου $\mathbb{X}$ κατασκευάζει απλότοπο για κάθε πεπερασμένο υποσύνολο του $\mathbb{X}$ με διάμετρο το πολύ α.*

Παράμετροι `RipsComplex`

- `distance_matrix`: ο πίνακας αποστάσεων
- `max_edge_length`: μέγιστη ακμή α
- `max_dimension`: μέγιστη διάσταση

Ραγδαία Αύξηση Πολυπλοκότητας!

Ο αριθμός απλοτόπων αυξάνεται **εκθετικά** με τον αριθμό σημείων και τη διάσταση. Για 370 σημεία και $\alpha = 0.8$ υπάρχουν **πάνω από** 10^6 **απλότοποι**!

Τιμή Φιλτραρίσματος

Η τιμή φιλτραρίσματος κάθε απλοτόπου είναι η **διάμετρός** του (μέγιστη απόσταση μεταξύ κορυφών).

Κώδικας: Rips από Πίνακα Αποστάσεων

Κατασκευή & Περιγραφή Rips Συμπλέγματος

```
# Κατασκευή σκελετού1- από πίνακα αποστάσεων
skeleton_protein = gd.RipsComplex(
distance_matrix = D.values,
max_edge_length = 0.8
)

# Επέκταση σε διάσταση 2
Rips_simplex_tree_protein = skeleton_protein.create_simplex_tree(max_dimension=2)

# Διάσταση: 2
print(Rips_simplex_tree_protein.dimension())

# Κορυφές: 370 όλα( τα σημεία)
print(Rips_simplex_tree_protein.num_vertices())

# Απλότοποι: > 1.000.000 !!
print(Rips_simplex_tree_protein.num_simplices())

```

Επανάληψη Φιλτραρίσματος (Iterating the Filtration)

```
rips_filtration = Rips_simplex_tree_protein.get_filtration()
rips_list = list(rips_filtration)
# Εκτύπωση 400 πρώτων στοιχείων
for splx in rips_list[0:400]:
print(splx)
```

Ερμηνεία Φιλτραρίσματος (Filtration Interpretation)

Ανάγνωση Αποτελεσμάτων `get_filtration()`

- Κάθε στοιχείο είναι της μορφής `([i,...,j], t)`: λίστα ακεραίων (δείκτες σημείων) και τιμή φιλτραρίσματος *t*
- Για κορυφές: τιμή φιλτραρίσματος $= 0$
- Η **πρώτη ακμή** στο φιλτράρισμα είναι `[289, 290]`: τα δύο πλησιέστερα σημεία, σε απόσταση 0.015

Σημασία της Σειράς Φιλτραρίσματος

Το φιλτράρισμα μας δίνει μια **χρονική εξέλιξη** της τοπολογικής δομής: από μεμονωμένα σημεία ($\alpha = 0$) έως πλήρως συνδεδεμένο σύμπλεγμα ($\alpha = 0.8$).

Alpha Σύμπλεγμα μέσω MDS (Alpha Complex via MDS)

Πρόβλημα

Το Alpha σύμπλεγμα απαιτεί **γεωμετρικές συντεταγμένες** στον $\mathbb{R}^p$ (για Delaunay τριγωνοποίηση), όχι μόνο πίνακα αποστάσεων.

Λύση: MDS (Multidimensional Scaling)

Η **Πολυδιάστατη Κλιμάκωση** (Multidimensional Scaling) βρίσκει αναπαράσταση n σημείων στον $\mathbb{R}^p$ που **διατηρεί κατά προσέγγιση** τις ζευγαρωτές αποστάσεις.

Ιδιότητες MDS

- Παράμετρος `n_components`: διάσταση εμβάπτισης
- Μέσω MDS ισχύει ότι `dissimilarity='precomputed'` δηλαδή γίνεται εισαγωγή έτοιμου πίνακα
- Αποτέλεσμα **προσεγγιστικό**: δεν διατηρεί πάντα ακριβώς τις αποστάσεις

Προσοχή

Το MDS παράγει μόνο **προσέγγιση** του αρχικού μετρικού χώρου.

Κώδικας: MDS & Alpha Σύμπλεγμα

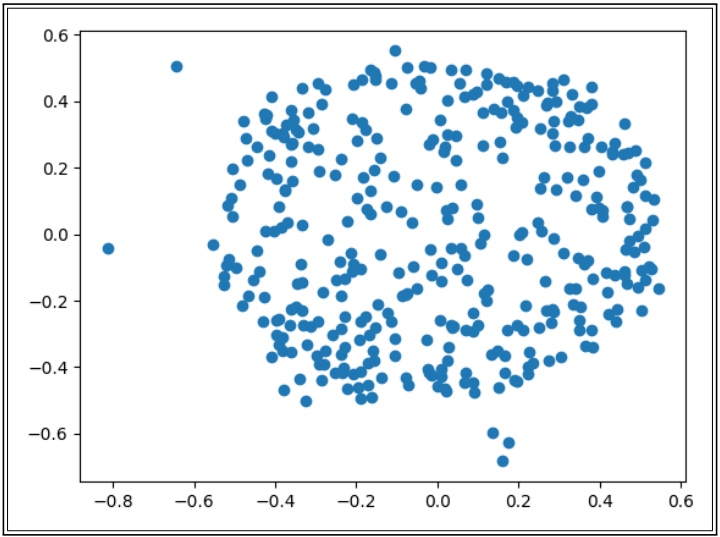
Εφαρμογή MDS στον Πίνακα Αποστάσεων

```
from sklearn.manifold import MDS

# Εμβάπτιση σε R^3 με MDS
embedding = MDS(n_components=3, dissimilarity='precomputed')
X_transformed = embedding.fit_transform(D)
print(X_transformed.shape)  # (370, 3)

# Οπτικοποίηση στις 2 πρώτες διαστάσεις
fig = plt.figure()
plt.scatter(X_transformed[:, 0], X_transformed[:, 1], label='MDS')

```

Κατασκευή Alpha Συμπλέγματος

```
# Κατασκευή Alpha Complex από τις συντεταγμένες MDS
alpha_complex = gd.AlphaComplex(points=X_transformed)
st_alpha = alpha_complex.create_simplex_tree()

print(st_alpha.dimension())      # 3
print(st_alpha.num_vertices())   # 370
print(st_alpha.num_simplices()) # 9255 -- ΠΟΛΥ λιγότεροι από Rips!

```

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.1: MDS & Σύμπλεγμα Alpha (Δύο Πρώτοι Άξονες)

Σύγκριση Rips vs Alpha (Rips vs Alpha Comparison)

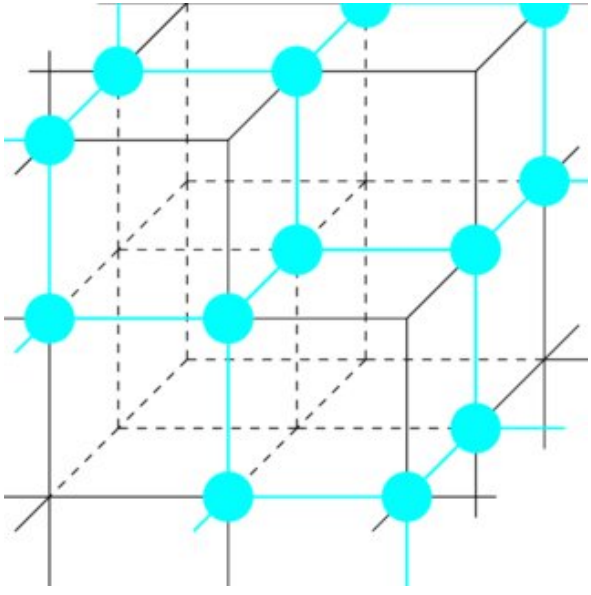
Αποτελέσματα για τα Δεδομένα MBP ($n = 370$ σημεία)

Ιδιότητα	**Rips ($\alpha = 0.8$)**	**Alpha (MDS)**
Κορυφές	370	370
Απλότοποι	$> 1.000.000$	9.255
Διάσταση	2	3
Χώρος εισόδου	Μετρικός	$\mathbb{R}^3$ (MDS)
Τύπος	Αφηρημένος	Γεωμετρικός

Κύριο Συμπέρασμα (Main Takeaway)

Το **Alpha σύμπλεγμα** παράγει **δραστικά λιγότερους** απλοτόπους (περίπου 100 φορές λιγότερους), καθιστώντας το υπολογιστικά πολύ πιο αποδοτικό – αλλά απαιτεί γεωμετρικές συντεταγμένες.

Πίνακας Περιεχομένων



Κυβικά Συμπλέγματα & Υποεπίπεδα Συνόλων

Εισαγωγή (Introduction)

Οι τοπολογικές υπογραφές μπορούν να εξαχθούν από τα **ανώτερα ή κατώτερα επίπεδα συνόλων** (upper/sublevel sets) μιας συνάρτησης $f : \mathbb{R}^p \mapsto \mathbb{R}$.

Μέθοδος Κυβικών Συμπλεγμάτων

- Ορισμός **κανονικού πλέγματος** (regular grid) στο $\mathbb{R}^d$
- Κορυφές στα σημεία του πλέγματος
- Η f παρέχει τις **τιμές φιλτραρίσματος** (filtration values)
- Εξέταση τοπολογίας υποεπίπεδων συνόλων (sublevel sets)

Βιβλιοθήκες

- `gudhi` – κυβικά συμπλέγματα
- `sklearn.neighbors` – `KernelDensity`
- `seaborn` – οπτικοποίηση πυκνότητας
- `numpy` – υπολογισμοί πλέγματος

Εισαγωγή Βιβλιοθηκών & Φόρτωση Δεδομένων

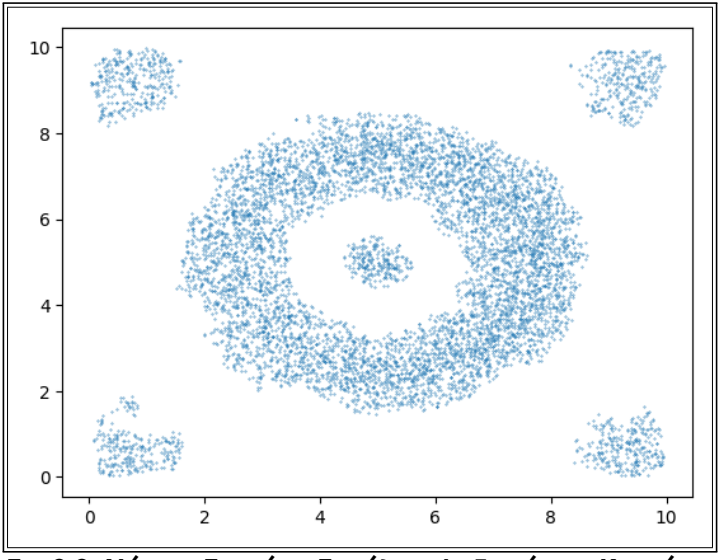
Imports

```
import numpy as np
import gudhi as gd
import pickle as pickle
from pylab import *
from sklearn.neighbors import KernelDensity
import seaborn as sns

```

Σύνολο Δεδομένων Κρατήρα (Crater Dataset)

```
# Φόρτωση νέφους σημείων κρατήρα
f = open("./datasets/crater_tuto", "rb")
crater = pickle.load(f)
f.close()

# Το νέφος ανήκει στο R^2:
# - κεντρικός δακτύλιος (central annulus)
# - τέσσερα σμήνη στις γωνίες (four corner clusters)
plt.scatter(crater[:, 0], crater[:, 1], s=0.1)

```

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.2: Νέφος Σημείων Συνόλου Δεδομένων Κρατήρα

Σύνολο Δεδομένων Κρατήρα (Crater Dataset)

Δομή του Νέφους Σημείων

Το νέφος βρίσκεται στο $\mathbb{R}^2$ και αποτελείται από:

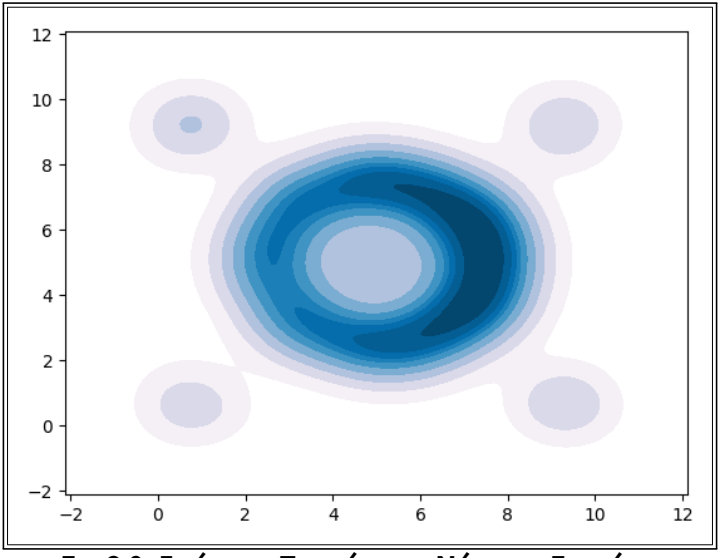
- Έναν **κεντρικό δακτύλιο** (central annulus) – τοπολογικό χαρακτηριστικό 1ης διάστασης (οπή)
- **Τέσσερα σμήνη** (clusters) στις γωνίες – τοπολογικά χαρακτηριστικά 0ης διάστασης (συνεκτικές συνιστώσες)

Στρατηγική Ανάλυσης

Αντί να κατασκευαστεί απευθείας Rips ή Alpha σύμπλεγμα, μελετάμε τα **ανώτερα επίπεδα συνόλων** (upper level sets) ενός **εκτιμητή πυκνότητας** (density estimator) ορισμένου στο νέφος. Οπτικοποίηση με `seaborn`:

```
sns.kdeplot(x=crater[:,0], y=crater[:,1], fill=True,
            cmap="PuBu", bw_method=.3)
```

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.3: Εκτίμηση Πυκνότητας Νέφους Σημείων

Κανονικό Πλέγμα & Εκτίμηση Πυκνότητας (Regular Grid & KDE)

Ορισμός Κανονικού Πλέγματος στο $[0, 10] \times [0, 10]$

```
xval = np.arange(0, 10, 0.05)  # 200 τιμές
yval = np.arange(0, 10, 0.05)  # 200 τιμές
nx = len(xval)   # 200
ny = len(yval)   # 200

```

Εκτίμηση Πυκνότητας Πυρήνα – KDE (Kernel Density Estimation)

```
# Εκπαίδευση KDE στα δεδομένα κρατήρα
kde = KernelDensity(kernel='gaussian', bandwidth=0.3).fit(crater)

# Υπολογισμός θέσεων πλέγματος
positions = np.array([[u, v] for u in xval for v in yval])

# Τιμές φιλτραρίσματος: αρνητική λογαριθμική πυκνότητα
# αρνητικό( πρόσημο: θέλουμε ανώτερα επίπεδα -> υποεπίπεδα)
filt_values = -kde.score_samples(X=positions)

print(min(filt_values), max(filt_values))

```

Λογική Αρνητικού Προσήμου (Negative Sign Logic)

Γιατί Αρνητική Λογαριθμική Πυκνότητα;

- Η `score_samples()` επιστρέφει **λογαριθμική πυκνότητα** (log-density), η οποία είναι πάντα **αρνητική**
- Το κυβικό σύμπλεγμα εξετάζει τα **υποεπίπεδα συνόλων** (sublevel sets) της συνάρτησης φιλτραρίσματος
- Για να μελετήσουμε τα **ανώτερα επίπεδα** (upper level sets) της πυκνότητας, παίρνουμε το **αρνητικό**: $f_{\text{filt}} = -\log(\hat{p})$
- Τα **υποεπίπεδα** του $-\log \hat{p}$ αντιστοιχούν στα **ανώτερα επίπεδα** της $\hat{p}$

Συνέπεια (Implication)

Τα **τοπολογικά χαρακτηριστικά** που ανακαλύπτουμε (συνεκτικές συνιστώσες, οπές) αντιστοιχούν σε **μέγιστα πυκνότητας** και αποκαλύπτουν τη δομή σμηνών του νέφους σημείων.

Κατασκευή Κυβικού Συμπλέγματος (Building the Cubical Complex)

Δημιουργία `CubicalComplex` με GUDHI

```
# Κατασκευή κυβικού συμπλέγματος
# Το filt_values αναδιαμορφώνεται σε πλέγμα [nx x ny]
cc_density_crater = gd.CubicalComplex(
top_dimensional_cells=filt_values.reshape([nx, ny])
)

# Τύπος αντικειμένου: ΟΧΙ SimplexTree!
print(type(cc_density_crater))
# <class 'gudhi.cubical_complex.CubicalComplex'>

# Παρόμοιες μέθοδοι με SimplexTree:
print(cc_density_crater.dimension())   # 2
print(cc_density_crater.num_simplices())

```

Δομή Κυβικού Συμπλέγματος (Cubical Complex Structure)

- **0-κύβοι** (0-cubes): κορυφές του πλέγματος
- **1-κύβοι** (1-cubes): ακμές πλέγματος (οριζόντιες & κατακόρυφες)
- **2-κύβοι** (2-cubes): τετράγωνα κελιά πλέγματος

Κυβικό vs Απλοτοπικό Σύμπλεγμα (Cubical vs Simplicial Complex)

Κυβικό Σύμπλεγμα (Cubical)

- Βασίζεται σε **ορθογώνιο πλέγμα** (rectangular grid)
- Κατάλληλο για **εικόνες** και **πλεγματικά δεδομένα**
- Η f ορίζεται στα **κελιά** (cells) του πλέγματος
- Κλάση: `CubicalComplex` (ΟΧΙ `SimplexTree`)

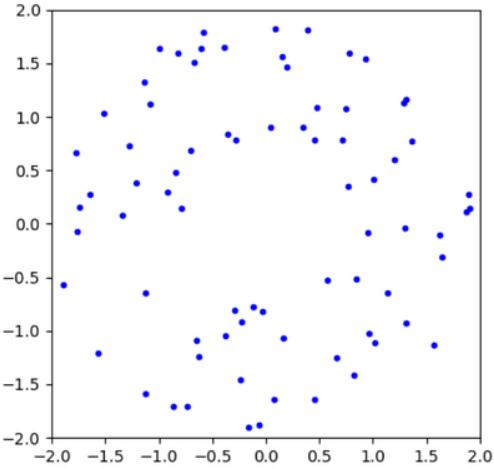
Απλοτοπικό Σύμπλεγμα (Simplicial)

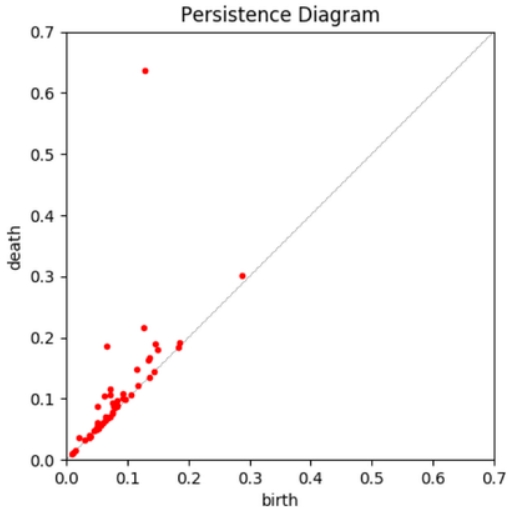
- Βασίζεται σε **νέφος σημείων** (point cloud)
- Κατάλληλο για **άμορφα** δεδομένα
- Η f ορίζεται στις **κορυφές**
- Κλάση: `SimplexTree`

Κοινό Χαρακτηριστικό

Και οι δύο κλάσεις παρέχουν τις ίδιες βασικές μεθόδους: `dimension()`, `num_simplices()`, `persistence()`, κλπ.

Πίνακας Περιεχομένων





Επίμονη Ομολογία & Διαγράμματα Επιμονής

Εισαγωγή

Η **Επίμονη Ομολογία** (Persistent Homology) είναι το κεντρικό εργαλείο της TDA: κωδικοποιεί την **εξέλιξη τοπολογικών χαρακτηριστικών** (οπές κ.λπ.) σε όλες τις κλίμακες του φιλτραρίσματος.

Βιβλιοθήκες

```
1       import numpy as np
2       import pandas as pd
3       import gudhi as gd
4       from sklearn import manifold
5       from pylab import *
6
```

Ομολογικές Ομάδες & Οπές (Homology Groups & Holes)

Κίνητρο (Motivation)

Δύο τοπολογικά σχήματα μπορούν να **διαχωριστούν** εξετάζοντας τις **οπές** τους (holes).

Διαστάσεις Οπών (Hole Dimensions)

- **Διάσταση 0**: συνεκτικές συνιστώσες (connected components)
- **Διάσταση 1**: βρόχοι / κυκλικές οπές (loops)
- **Διάσταση 2**: κοιλότητες (voids)
- **Διάσταση** k: k-διάστατες οπές

Επίμονη Ομολογία (Persistent Homology)

Παρακολουθεί τη **γέννηση** (birth) και **θάνατο** (death) κάθε τοπολογικού χαρακτηριστικού καθώς το φίλτρο α μεγαλώνει.

Επιμονή (Persistence)

$$\text{persistence} = \text{death} - \text{birth}$$

Μεγάλη επιμονή $\Rightarrow$ σημαντικό χαρακτηριστικό!

Εξέλιξη Ένωσης Σφαιρών (Evolution of Union of Balls)

Διαισθητική Εικόνα (Intuitive Picture)

Εξετάζουμε ένωση σφαιρών αυξανόμενης ακτίνας r με κέντρα τα σημεία C:

(α) $r = 0$: μεμονωμένα σημεία – κάθε σημείο είναι ξεχωριστή συνεκτική συνιστώσα (0-διάστατο χαρακτηριστικό)

(β) Οι σφαίρες αρχίζουν να **επικαλύπτονται**: ορισμένες συνεκτικές συνιστώσες **συγχωνεύονται** (θάνατος χαρακτηριστικών 0 διάστασης)

(γ) Ενιαία συνεκτική συνιστώσα – εμφάνιση **1-διάστατων κύκλων** (βρόχοι, γέννηση μπλε διαστημάτων)

(δ) Ένας κύκλος **γεμίζει**: θάνατος 1-διάστατου χαρακτηριστικού

(ε) Παραμένει το **ανθεκτικό-επίμονο** 1-διάστατο χαρακτηριστικό (κόκκινο διάστημα)

Γραμμοκώδικες & Διάγραμμα Επιμονής (Barcode & Persistence Diagram)

Ορισμός (Definition)

Ο **γραμμοκώδικας** (barcode) αναπαριστά κάθε τοπολογικό χαρακτηριστικό ως ένα διάστημα $[b, d)$ όπου b = γέννηση (birth), d = θάνατος (death).

Ερμηνεία

- Μεγάλο μήκος $\Leftrightarrow$ **σημαντικό** χαρακτηριστικό
- Μικρό μήκος $\Leftrightarrow$ τοπολογικός **θόρυβος**

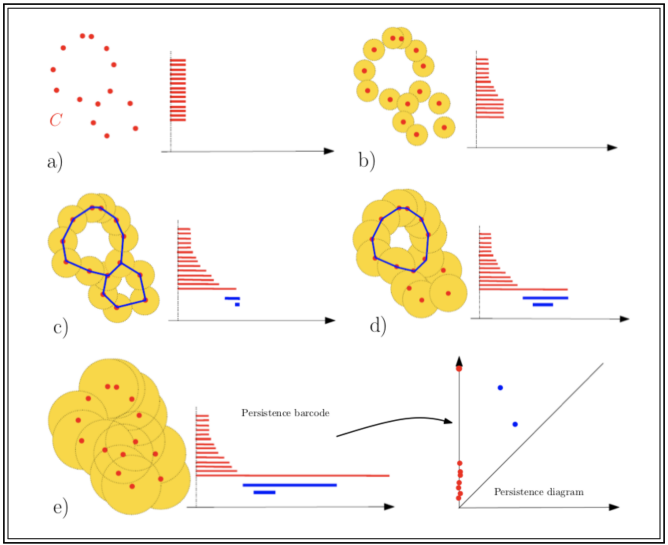
Ορισμός (Definition)

Το **διάγραμμα επιμονής** (persistence diagram) αναπαριστά κάθε χαρακτηριστικό ως σημείο (b, d) στο $\mathbb{R}^2$.

Ερμηνεία

- Σημεία **μακριά από τη διαγώνιο** $\Leftrightarrow$ σημαντικά χαρακτηριστικά
- Σημεία **κοντά στη διαγώνιο** $\Leftrightarrow$ θόρυβος

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.4: Γραμμοκώδικας & Διάγραμμα Επιμονής

Υπολογισμός Επίμονης Ομολογίας στο Rips Σύμπλεγμα

Κατασκευή Rips & Υπολογισμός Ομολογίας

```
# Κατασκευή Rips από τον πρώτο πίνακα αποστάσεων
D0 = dist_list[0]

skeleton_protein0 = gd.RipsComplex(
distance_matrix=D0.values,
max_edge_length=0.8
)
Rips_simplex_tree_protein0 = skeleton_protein0.create_simplex_tree(max_dimension=2)

# Υπολογισμός επίμονης ομολογίας
BarCodes_Rips0 = Rips_simplex_tree_protein0.persistence()

# Εκτύπωση 20 πρώτων στοιχείων γραμμοκώδικα
for i in range(20):
print(BarCodes_Rips0[i])
# Αποτέλεσμα: (1, (b, d)) -- διάστατα1- χαρακτηριστικά κύκλοι()

```

Δομή `BarCodes_Rips0`

Κάθε στοιχείο: `(dim, (birth, death))` – διάσταση, γέννηση, θάνατος

Ανάλυση Διαστημάτων Επιμονής (Persistence Intervals Analysis)

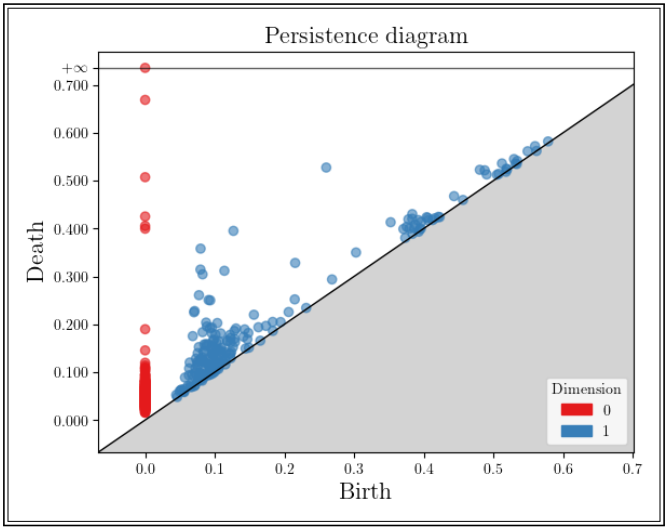
Διαστήματα ανά Διάσταση (Intervals per Dimension)

```
# Διαστήματα επιμονής για διάσταση 0 συνεκτικές( συνιστώσες)
Rips_simplex_tree_protein0.persistence_intervals_in_dimension(0)
# Το τελευταίο διάστημα (0.0, inf) -- ποτέ δεν πεθαίνει!

# Οπτικοποίηση διαγράμματος επιμονής
gd.plot_persistence_diagram(BarCodes_Rips0)

```

Ερμηνεία Διαγράμματος Επιμονής

- **Κόκκινα σημεία** (0-διάσταση): συνεκτικές συνιστώσες – το τελευταίο στο ∞
- **Μπλε σημεία** (1-διάσταση): κύκλοι/βρόχοι
- Σημεία **μακριά από τη διαγώνιο**: σημαντικές τοπολογικές δομές της MBP
- Σημεία **κοντά στη διαγώνιο**: τοπολογικός θόρυβος

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.5: Διάγραμμα Επιμονής Γεννήσεων & Θανάτων

Απόσταση Bottleneck (Bottleneck Distance)

Ορισμός (Definition)

Ένα **ταίριασμα** (matching) m μεταξύ δύο διαγραμμάτων $\mathrm{dgm}_1, \mathrm{dgm}_2$ είναι υποσύνολο του $\mathrm{dgm}_1 \times \mathrm{dgm}_2$ ώστε κάθε σημείο εκτός διαγωνίου να εμφανίζεται ακριβώς μία φορά. Η **απόσταση Bottleneck** ορίζεται ως:

$$d_b(\mathrm{dgm}_1, \mathrm{dgm}_2) = \inf_m \max_{(p,q)\in m} \|p - q\|_\infty$$

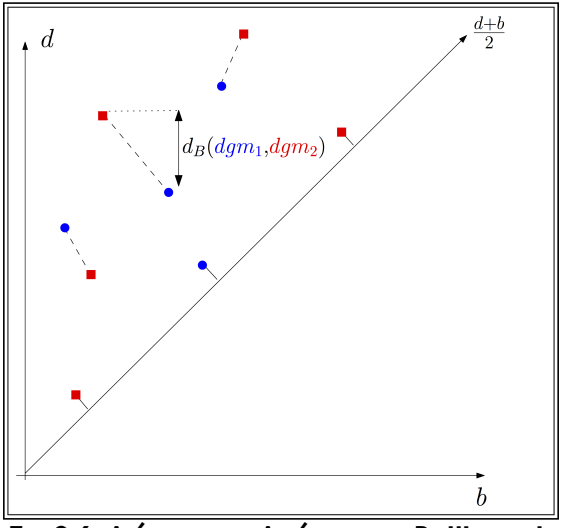
Σταθερότητα (Stability)

Η απόσταση Bottleneck είναι σταθερή: μικρές διαταραχές δεδομένων $\Rightarrow$ μικρή αλλαγή στο διάγραμμα.

Χρήση

- Σύγκριση τοπολογικών υπογραφών
- Ταξινόμηση διαμορφώσεων MBP
- Θεωρητική βάση για στατιστικές μεθόδους

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.6: Διάγραμμα Απόστασης Bottleneck

Κώδικας: Απόσταση Bottleneck

Υπολογισμός Rips για 2η Διαμόρφωση

```
    D1 = dist_list[1]
    skeleton_protein1 = gd.RipsComplex(
    distance_matrix=D1.values, max_edge_length=0.8
    )
    Rips_simplex_tree_protein1 = skeleton_protein1.create_simplex_tree(max_dimension=2)
    BarCodes_Rips1 = Rips_simplex_tree_protein1.persistence()

```

Υπολογισμός Απόστασης Bottleneck

```
    # Διαστήματα διάστασης 1 για τις δύο διαμορφώσεις
    I0 = Rips_simplex_tree_protein0.persistence_intervals_in_dimension(1)
    I1 = Rips_simplex_tree_protein1.persistence_intervals_in_dimension(1)

    # Ακριβής υπολογισμός αργός()
    gd.bottleneck_distance(I0, I1)

    # Προσεγγιστικός υπολογισμός γρήγορος(, σφάλμα <= 0.01)
    gd.bottleneck_distance(I0, I1, 0.01)

```

Σημείωση

Ο προσεγγιστικός υπολογισμός είναι συνήθως **πολύ ταχύτερος** από τον ακριβή.

MDS στις Αποστάσεις Bottleneck (MDS on Bottleneck Distances)

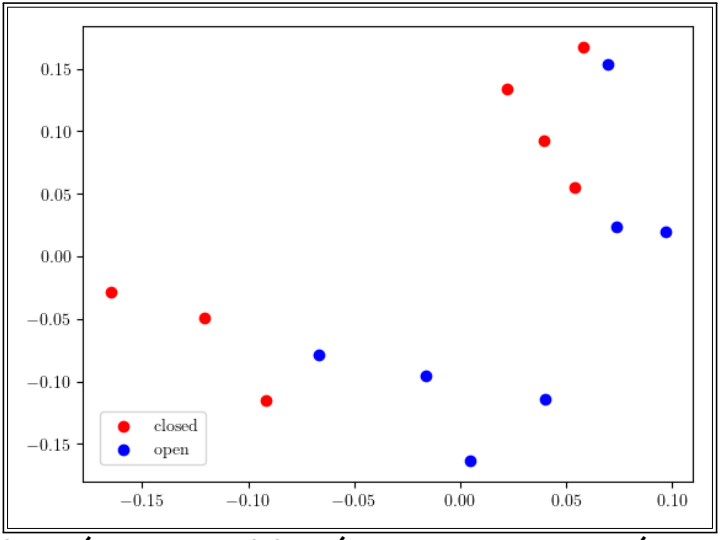
Υπολογισμός Πινάκων Bottleneck για όλες τις Διαμορφώσεις

```
    persistence_list0 = []  # Διαστήματα διάστασης 0
    persistence_list1 = []  # Διαστήματα διάστασης 1

    for d in dist_list:
    rips = gd.RipsComplex(distance_matrix=d.values, max_edge_length=0.8)
    st = rips.create_simplex_tree(max_dimension=2)
    st.persistence()
    persistence_list0.append(st.persistence_intervals_in_dimension(0))
    persistence_list1.append(st.persistence_intervals_in_dimension(1))

    # Κατασκευή πινάκων αποστάσεων 14x14
    l = len(files_list)
    B0, B1 = np.zeros((l, l)), np.zeros((l, l))
    for i in range(l):
    for j in range(i):
    B0[i,j] = gd.bottleneck_distance(persistence_list0[i], persistence_list0[j])
    B1[i,j] = gd.bottleneck_distance(persistence_list1[i], persistence_list1[j])
    B0 = B0 + B0.transpose()
    B1 = B1 + B1.transpose()
```

Εφαρμογή MDS στους Πίνακες Bottleneck

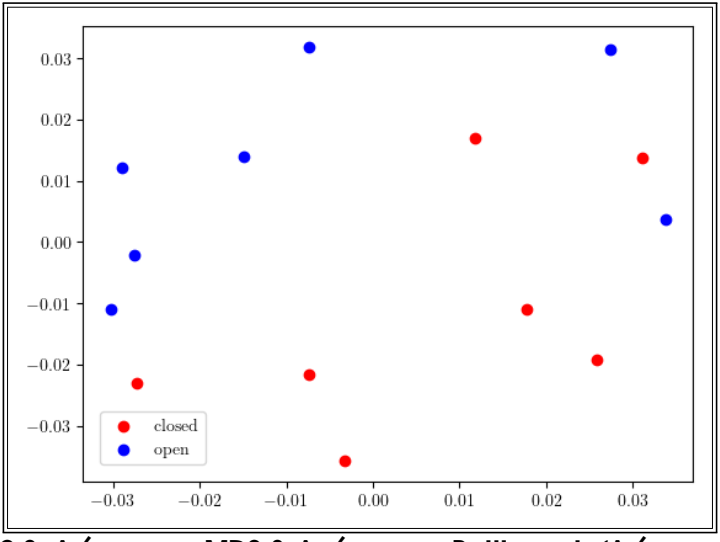
```
    mds = manifold.MDS(
    n_components=2, max_iter=3000, eps=1e-9,
    dissimilarity="precomputed", n_jobs=1
    )

    # MDS βάσει απόστασης Bottleneck διάσταση( 0)
    pos = mds.fit(B0).embedding_
    plt.scatter(pos[0:7,  0], pos[0:7,  1], color='red',  label="closed")
    plt.scatter(pos[7:1,  0], pos[7:1,  1], color='blue', label="open")
    plt.legend(loc=3, borderaxespad=1)

    # MDS βάσει απόστασης Bottleneck διάσταση( 1)
    pos = mds.fit(B1).embedding_
    plt.scatter(pos[0:7,  0], pos[0:7,  1], color='red',  label="closed")
    plt.scatter(pos[7:1,  0], pos[7:1,  1], color='blue', label="open")
    plt.legend(loc=3, borderaxespad=1)

```

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.7: Διάγραμμα MDS & Απόστασης Bottleneck (Διάσταση 0)

Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 2.8: Διάγραμμα MDS & Απόστασης Bottleneck (Διάσταση 1)

Αποτέλεσμα: Διαχωρισμός Ανοιχτής/Κλειστής Μορφής MBP

Ερμηνεία Αποτελέσματος MDS

- Οι **κλειστές** (closed) διαμορφώσεις (κόκκινο) ομαδοποιούνται μαζί στον MDS χώρο
- Οι **ανοιχτές** (open) διαμορφώσεις (μπλε) σχηματίζουν ξεχωριστή ομάδα
- Η **τοπολογική υπογραφή** (topological signature) της επίμονης ομολογίας **διακρίνει** τις δύο λειτουργικές καταστάσεις της MBP

Κύριο Συμπέρασμα (Main Takeaway)

Η TDA με επίμονη ομολογία και αποστάσεις Bottleneck αποκαλύπτει **βιολογικά σημαντικές** δομικές διαφορές μεταξύ ανοιχτής και κλειστής μορφής πρωτεΐνης – χωρίς να χρησιμοποιηθούν γεωμετρικές αποστάσεις $\mathbb{R}^3$, αλλά **μόνο δυναμικές συσχετίσεις**.

Πίνακας Περιεχομένων

Σύνοψη Εννοιών (Summary of Concepts)

Ενότητα 1: Συμπλέγματα από Πίνακες Αποστάσεων

- **Rips** από `distance_matrix` – δυναμικές αποστάσεις MBP
- **MDS** για εμβάπτιση μετρικού χώρου σε $\mathbb{R}^3$
- **Alpha** από MDS συντεταγμένες – πολύ λιγότεροι απλότοποι

Ενότητα 2: Κυβικά Συμπλέγματα

- **CubicalComplex** – για πλεγματικά δεδομένα & εικόνες
- **KDE** ως συνάρτηση φιλτραρίσματος
- Αρνητικό πρόσημο: υποεπίπεδα $\Leftrightarrow$ ανώτερα επίπεδα πυκνότητας

Ενότητα 3: Επίμονη Ομολογία

- `persistence()`: γραμμοκώδικας (barcode) & διάγραμμα επιμονής
- **Απόσταση Bottleneck** – σύγκριση τοπολογικών υπογραφών
- **MDS** στις αποστάσεις Bottleneck – ταξινόμηση MBP

Σας ευχαριστώ θερμά για την προσοχή σας!

Ερωτήσεις;

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS