КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики
Кафедра обчислювальної математики

Курсова робота
за спеціальністю 113 «Прикладна математика»
на тему:

ГІПОТЕЗА ПРО КОМПАКТНІСТЬ ФРАКТАЛЬНОЇ РОЗМІРНОСТІ ЯДЕР БУКАЛЬНОГО ЕПІТЕЛІЮ У ХВОРИХ НА РАК МОЛОЧНОЇ ЗАЛОЗИ

Виконала студентка 3-го курсу
ОПП «Прикладна математика»
Чопа Ярослава Іванівна _____
(підпис)

Науковий керівник:
професор кафедри обчислювальної математики,
доктор фізико-математичних наук, професор
Клюшин Дмитро Анатолійович _____
(підпис)

Засвідчую, що в цій роботі
немає запозичень із праць інших авторів
без відповідних посилань.
Студентка _____
(підпис)

КИЇВ 2025

РЕФЕРАТ

Обсяг роботи: 21 сторінка основного тексту, 15 джерел літератури, 1 додаток.

Ключові слова: КЛАСТЕРИЗАЦІЯ, КОМПАКТНІСТЬ, МІРА МІНЬКОВСЬКОГО, МІРА СХОЖОСТІ, *P*-СТАТИСТИКА, СПЕЦИФІЧНІСТЬ, ФРАКТАЛЬНА РОЗМІРНІСТЬ, ЧУТЛИВІСТЬ.

Об'єкт дослідження. Фрактальні розмірності контурів ядер букального епітелію, отримані від пацієнток із раком молочної залози та від здорових донорів контрольної групи.

Мета роботи. Перевірити гіпотезу про статистичну компактність розподілу фрактальних розмірностей у групі *Breast Cancer* порівняно з контрольною вибіркою та оцінити придатність цієї ознаки для подальшої автоматизованої діагностики раку молочної залози.

Методи та інструменти дослідження.

- обчислення фрактальної (Мінковської) розмірності методом *box–counting*;
- розрахунок симетричної міри схожості на основі *p*-статистики (довірчі межі Вілсона, $g{=}3$);
- побудова матриць схожості та їх кластеризація алгоритмами Spectral, Agglomerative k-Means і GMM;
- програмна реалізація в Python 3.12 (NumPy, SciPy, scikit-learn, Matplotlib).

Основні результати.

1. Сформовано вибірку з 322 цифрових сканограм (204 — *BC*, 118 — *Control*) у трьох спектральних каналах (Blue, Green, Red).
2. Для кожної пари сканограм обчислено *p*-статистику, що дала чотири повні матриці схожості (загальну та три каналоспецифічні).
3. Двокластерний аналіз показав, що в зеленому каналі Spectral Clustering досягає чутливості 0.868 та специфічності 0.966, порівняно з нижчими показниками для інших каналів та методів.
4. Встановлено статистично значущу компактність ядрових розмірностей у групі *BC* та часткове перекриття контролю з межовими зонами розподілу.

Практичне значення результатів. Доведена компактність ознаки створює підґрунтя для побудови моделі штучного інтелекту, що, на основі лише мазка бу-

кального епітелію, зможе автоматично класифікувати зразки та потенційно виявляти рак молочної залози на ранніх стадіях без інвазивних процедур.

Рекомендації щодо впровадження. Планується:

- розширити вибірку та залучити дані з незалежних центрів;
- використати зелений канал як базове представлення у майбутній нейромережевій моделі;
- інтегрувати *p*-статистику як ядро схожості в напівавідірвані або глибокі кластеризатори для підвищення діагностичної точності.

Зміст

ВСТУП

Букальний епітелій — шар плоских клітин, що вистеляє внутрішню поверхню щоки, — давно визнається зручним маркером системних фізіологічних змін. Протягом останніх трьох десятиліть численні цитологічні дослідження засвідчили, що злоякісні процеси, зокрема рак молочної залози, супроводжуються вимірюваними змінами ядерної морфології, які можна зафіксувати зі стандартних мазків і фотомікрографій. Одним із кількісних дескрипторів, що стабільно виявляє такі зміни, є *фрактальна (Мінковського) розмірність* контуру ядра, оцінена тут алгоритмом box–counting. Попри зростання обсягу свідчень, що різні патологічні стани відповідають характерним розподілам фрактальних розмірностей ядер, статистична компактність цих розподілів і їхня придатність для некерованої стратифікації пацієнтів залишаються недостатньо з'ясованими.

У цій роботі ці питання досліджуються для раку молочної залози. Сформульовано та перевіряється така гіпотеза компактності: «Фрактальні розмірності ядер букального епітелію у пацієнток із раком молочної залози утворюють розподіл, який є статистично відмінним від відповідного розподілу для здорових жінок і водночас внутрішньо більш однорідним».

Мета та завдання. Метою курсової роботи є перевірка зазначеної гіпотези на ретельно відібраній вибірці оцифрованих цитологічних зображень. Для досягнення цієї мети розв'язуються чотири конкретні завдання:

— побудувати, незалежно для кожного RGB-каналу, матриці подібності на основі p-статистики, що кількісно характеризують внутрішньогрупову (*Breast Cancer*, *Control*) та міжгрупову подібність фрактальних розмірностей ядер;
— дослідити вплив параметра g p-статистики на чисельну точність і діагностичну роздільність;
— застосувати кілька класичних методів кластеризації (ієрархічне зв'язування, k-means і гаусівські змішані моделі) до отриманих матриць подібності та порівняти отримані розбиття пацієнтів;
— інтерпретувати результати кластеризації відносно клінічних міток і проаналізувати їхню стабільність між кольоровими каналами.

Об'єкт, методи та інструменти. Об'єктом дослідження є вибірка значень фрактальної розмірності (за методом box–counting) для 68 пацієнток із раком молочної залози та 29 здорових жінок з контрольної групи. Кожна пацієнтка надає

три файли — по одному на RGB-канал — кожен із 51 значенням фрактальної розмірності ядра. Методологічно дослідження поєднує: (i) обчислення мінковських розмірностей методом box–counting; (ii) p-статистику як міру компактності вибірок та попарної подібності; і (iii) некеровані алгоритми машинного навчання для виявлення кластерів. Усі обчислення виконано в Python 3.12 із використанням пакетів NumPy, SciPy та Scikit-learn; візуалізації створено за допомогою Matplotlib і Seaborn.

Очікувана застосовність. Надійний неінвазивний маркер, здатний розрізняти здорові та злоякісні системні стани, може доповнити чинні програми скринінгу, знизити потребу в інвазивних біопсіях і бути поширений на інші онкологічні та системні захворювання. Крім того, стійка кластеризація пацієнтів на основі лише мазків букального епітелію може полегшити масштабні епідеміологічні дослідження та сприяти персоналізованому плануванню лікування.

Зв'язок із попередніми дослідженнями. Курсова робота спирається на фундаментальні дослідження, виконані на кафедрі біомедичної фізики Київського національного університету імені Тараса Шевченка, де вперше було встановлено зв'язок між фрактальною морфологією ядер і онкопатологією. Робота також перегукується з міжнародними дослідженнями, що застосовують фрактальну геометрію та аналіз текстур для діагностики раку. Запропонована методологія узагальнює попередні двовибіркові порівняння, інтегруючи міжканальний аналіз і сучасні кластеризаційні конвеєри.

1 ПОПЕРЕДНЯ ОБРОБКА ДАНИХ

1.1. Група пацієнток та підготовка зразків

У дослідженні взяли участь $N = 130$ доброволиць, розподілених на три діагностичні групи: 68 пацієнток із раком молочної залози (BC), 33 пацієнтки з фіброаденоматозом (FA) та 29 здорових донорок (HC). Зразки букального епітелію отримували після полоскання ротової порожнини та м’якого видалення поверхневого шару клітин. Зіскрібки брали з двох звичних глибин шипуватого шару (позначених “median” і “deep”), висушували на повітрі та фіксували протягом 30 хв у суміші Нікіфорова. ДНК фарбували за реакцією Фельґена після холодного гідролізу в 5 *N* HCl протягом 15 хв при 21–22°C.

Цитоспектрофотометрію виконували на мікроскопі *Olympus* BX, оснащеному цифровою камерою *Camedia C-5050* та спеціалізованим програмним забезпеченням. Для кожного мазка аналізували в середньому 52 ± 4 інтерфазні ядра, що дало загалом 20 256 зображень ядер. Кожне ядро фотографували за трьох оптичних умов — без фільтра, з жовтим та з фіолетовим фільтрами — отримуючи 6752 зображення на кожну умову. Сирі мікрофотографії містять три кольорові канали (R, G, B) та відповідне сірошкальне відображення. Вміст ДНК–фуксину визначали як добуток середньої оптичної густини на проєктовану площу ядра; просторовий розподіл густини зберігали у вигляді матриці 160×160 пікселів, що надалі називається *ДНК-сканограмою*.

1.2. Попередня обробка зображень

Для кожної пацієнтки фіксують від 23 до 81 ядер. З кожного ядра одержують три сканограми (без фільтра, жовтий, фіолетовий), кожну — у чотирьох версіях (три кольорові канали та сірошкала). Попередні експерименти показали, що синій канал, отриманий через жовтий фільтр, майже насичений (темно-сірий до чорного) й тому виключений із подальшого аналізу. Усі решта комбінацій «фільтр–канал» надходять на стадію попередньої обробки.

Мікроскопічні зображення чутливі до шуму датчика, артефактів фарбування та сторонніх часток, що потребує усунення перед кількісним аналізом. Достатньою виявилася така послідовність.

Крок 1: Фільтрація шуму. Кожен канал обробляли медіанним фільтром 5×5, що зменшує імпульсний шум, зберігаючи контури ядер.

Крок 2: Бінаризація. Глобальний поріг Отсу перетворював сірошкальне зображення на бінарну маску,де передній план відповідає ядерному матеріалу.

Крок 3: Морфологічне очищення. Залишковий шум *«сіль-і-перець»* усували операцією *відкривання* (ерозія, за якою дилатація) для видалення ізольованих темних пікселів, після чого *закривання* заповнювало дрібні світлі пори.

Крок 4: Виділення ознак.Фрактальну складність контуру кожного ядра кількісно оцінювали за допомогою Мінковської (метод підрахунку квадратів) розмірності, використовуючи модифікований багатомасштабний алгоритм.

Крок 5: Плоско-польова (flat-field) корекція. Для кожного сеансу мікроскопування записували зображення «порожнього кадру» - рівномірно освітлену ділянку без препаратів. Подальше покоміркове ділення робочого кадру на середньозважений flat-field вирівнювало неосвітленість поля та усувало віньєтування по краях поля зору.

Крок 6: Нормалізація динамічного діапазону. Після корекції фону гістограму інтенсивностей кожного каналу лінійно розтягували у межі $[0, 255]$ згідно з $1^{\text{м}}$ та $99^{\text{м}}$ перцентилями, щоб компенсувати різницю експозицій та забезпечити сталу контрастність.

Крок 7: Стандартизація масштабу. Оскільки розмір ядра на знімку залежить від геометрії мікроскопа та положення предметного скла, усі бінарні маски масштабували афінним перетворенням до фіксованого еквівалентного діаметра d_0 (в пікселях). Це дозволяє порівнювати фрактальність контурів різних об'єктів без впливу абсолютного розміру.

Крок 8: Уніфікація орієнтації. Контури ядер центрували та повертали так, щоб головна вісь еліптичної апроксимації збігалася з горизонталлю кадру; метод головних компонент гарантує однозначність повороту. Процедура зменшує ефект рандомної орієнтації ядра на фрактальні коефіцієнти, зберігаючи форму.

Крок 9: Баланс білого для кольорових каналів. Для RGB-версій застосовувався простий grey-world баланс: середнє значення по кожному каналу приводили до спільного середнього, щоб нейтралізувати відтінки, внесені джерелом освітлення та оптикою мікроскопа.

2 МІРА СХОЖОСТІ ТА P-СТАТИСТИКА

2.1. Огляд непараметричних методів

Непараметричні статистичні процедури забезпечують висновування без припущень щодо конкретної параметричної форми базового розподілу. Замість параметрів (наприклад, середніх чи дисперсій) таких розподілів, непараметричні тести часто використовують ранги або емпіричний розподіл даних. Класичними прикладами є двовибірковий критерій Колмогорова–Смирнова для перевірки рівності розподілів та ранговий тест Вілкоксона для порівняння медіан, які роблять мінімальні припущення і мають нульові розподіли, що не залежать від параметрів. Такі методи корисні для перевірки однорідності двох вибірок без зазначення параметричної моделі. Однак стандартні непараметричні тести можуть мати обмежену потужність проти певних альтернатив (наприклад, різниця у варіативності за близьких локалізацій або навпаки). У таких випадках доцільно застосовувати спеціалізовані міри близькості вибірок.

Одним із потужних підходів є безпосереднє вимірювання *схожості* чи *однорідності* двох вибірок шляхом порівняння їх емпіричних розподілів. У роботах Клюшина та Петуніна було введено статистику, названу *p-статистикою*, що слугує мірою близькості вибірок та основою непараметричного тесту еквівалентності популяцій. p-статистика оцінює ймовірність того, що дві вибірки походять з одного розподілу (нульова гіпотеза однорідності) . На відміну від метрик відстані, які потребують метричного простору ознак, p-статистика працює безпосередньо з порядковими статистиками вибірок і використовує довірчі інтервали для біноміальних ймовірностей, як описано нижче. Показано, що вона ефективно виявляє як зсуви локації, так і масштабні відмінності між розподілами, навіть тоді, коли критерії Колмогорова–Смирнова чи Вілкоксона мають низьку потужність.

2.2. Довірчі інтервали та асимптотичні наближення

Перш ніж визначити p-статистику, розглянемо побудову довірчих інтервалів для часток, що є ключовим компонентом методу. *Довірчий інтервал* (ДІ) для невідомого параметра θ — це випадковий інтервал (L, U), побудований на основі

вибірки, який з імовірністю $1-\beta$ містить істинне значення параметра:

$$P_\theta\{L \le \theta \le U\} = 1-\beta. \tag{2.1}$$

Величина $1-\beta$ називається довірчим коефіцієнтом, а β — рівнем значущості (ймовірність того, що інтервал не покриває θ).

На практиці умову (2.1) виконати точно складно, тому використовують або точні, але консервативні, інтервали, або наближені інтервали, основані на великій вибірці. Наприклад, якщо $\hat{\theta}$ — незміщена оцінка θ з приблизною стандартною похибкою $\sigma_{\hat{\theta}}$, асимптотичний ДІ рівня $1-\beta$ можна отримати через нормальне наближення:

$$\hat{\theta} \pm z_{\beta/2}\,\sigma_{\hat{\theta}}, \tag{2.2}$$

де $z_{\beta/2}$ — квантиль стандартного нормального розподілу. Такий інтервал (інтервал Валда) для біноміальної частки відомий поганим покриттям, особливо за малих n або p, близьких до 0 чи 1. Надійнішим є інтервал Вілсона, який має кращі властивості покриття. У контексті p-статистики саме інтервал Вілсона використовується для перевірки відхилень спостережуваної частки від номінального значення.

2.3. Визначення та виведення p-статистики

Далі викладемо виведення p-статистики за підходом Клюшина та Петуніна. Розглянемо дві незалежні вибірки

$$x = (x_1, \dots, x_n) \sim G_1, \qquad y = (y_1, \dots, y_n) \sim G_2,$$

отримані з розподілів F_1 та F_2. Припустимо $n = m$ і що F_1, F_2 неперервні. Нехай $x_{(1)} \le \dots \le x_{(n)}$ — порядкові статистики x. Вони ділять вісь на $n+1$ інтервал, $(x_{(i)}, x_{(j)})$, $0 \le i < j \le n+1$, з $x_{(0)} = -\infty$, $x_{(n+1)} = \infty$. Маємо $N = \binom{n+1}{2}$ інтервалів.

За нульової гіпотези $H_0 : F_1 = F_2$

$$P\{y_k \in (x_{(i)}, x_{(j)})\} = \frac{j-i}{n+1} \tag{2.3}$$

Нехай k_{ij} — кількість елементів y у цьому інтервалі, тоді

$$\hat{p}_{ij} = \frac{k_{ij}}{n}. \tag{2.4}$$

Вважаємо $k_{ij} \sim \text{Binomial}\big(n, \frac{j-i}{n+1}\big)$ і будуємо ДІ Вілсона $I_{ij}^{(n)}$ рівня $1-\beta$. Позначимо

$$I\Big\{\tfrac{j-i}{n+1} \in I_{ij}^{(n)}\Big\} = \begin{cases} 1, & \text{якщо } \frac{j-i}{n+1} \in I_{ij}^{(n)}, \\ 0, & \text{інакше.} \end{cases} \tag{2.5}$$

p-статистика визначається як

$$h = \mu(x, y) = \frac{1}{N} \sum_{0 \le i < j \le n+1} I\Big\{\tfrac{j-i}{n+1} \in I_{ij}^{(n)}\Big\}. \tag{2.6}$$

Таким чином h — це частка інтервалів, у яких спостережувана частка y узгоджується з теоретичною з імовірністю $1-\beta$. Отже, h є вибірковою оцінкою ймовірності того, що x і y взято з одного розподілу.

2.4. Асимптотичні властивості та довірчі межі

p-статистику використовують для тестування $H_0 : F_1 = F_2$ проти $H_1 : F_1 \neq F_2$. Правило: прийняти H_0, якщо h незначно менше $1-\beta$; відхилити, якщо h суттєво нижче. Для $1-\beta = 0.95$ відхиляємо H_0, якщо $h < 0.95$. Можна також побудувати ДІ для $p(B) = 1-\beta$ на основі спостережуваного h, трактуючи $L = Nh$ як біноміальну випадкову величину. Показано, що за $n \to \infty$

$$P_{H_0}\{h < 0.95\} \to 0.05, \tag{2.7}$$

тобто рівень помилки I роду асимптотично дорівнює β. Це обґрунтовує використання порогу 0.95.

Залежність подій у (2.6) не є строгою незалежністю, проте зі зростанням n вона слабшає, утворюючи *узагальнену схему Бернуллі*. Теореми про порядкові статистики гарантують асимптотичну узгодженість критерію.

Вибір параметра довіри $g = z_{\beta/2}$ заслуговує обговорення. Замість $z = 1.96$ інколи беруть $g = 3$, що забезпечує $\geq 95\%$ покриття для будь-якого унімодального розподілу завдяки нерівності Височанського–Петуніна. Такий вибір зберігає або

знижує рівень $\alpha \approx 0.05$, хоча й зменшує потужність.

2.5. p-статистика як міра схожості

Окрім тестування гіпотез, $h = \mu(x, y)$ є кількісною мірою схожості між вибірками й належить інтервалу $[0, 1]$. Значення $h \approx 1$ свідчить про високу схожість, $h \approx 0$ — про суттєві відмінності. На відміну від інтегральних відстаней (Колмогорова–Смирнова, L^2 тощо), p-статистика фіксує локальні розбіжності у кожному інтервалі порядкових статистик, що підвищує чутливість до змін форми розподілу. Завдяки незалежності від метрики, p-статистика лишається коректною для довільних неперервних розподілів значень. Вона поєднує функції критерію однорідності та індексу схожості, що робить її корисною у кластеризації, класифікації та детекції змін без параметричних припущень.

3 КЛАСТЕРИЗАЦІЯ. ПОНЯТТЯ ТА ОСНОВИ

Кластеризація — це метод неконтрольованого навчання, що розбиває множину точок даних на групи (кластери) таким чином, щоб елементи в одному кластері були більш схожими між собою, ніж із точками з інших кластерів. Властивості високої внутрішньокластерної схожості та низької міжкластерної схожості лежать в основі виявлення природних структур у даних. Кластеризація має широке теоретичне значення для задач, таких як сегментація зображень (групування пікселів у однорідні області) та класифікація невідімічених зразків, де внутрішні закономірності потрібно визначати без нагляду. Формально, маючи набір $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ у метричному просторі, кластеризацію можна визначити як розбиття $\mathcal{C} = \{C_1, \ldots, C_K\}$ множини $\{1, 2, \ldots, n\}$ (або, еквівалентно, X) на K неперетинних підмножин. Кожен кластер C_k репрезентує групу точок даних, визнаних схожими за певним критерієм (наприклад, відстанню чи густиною). Метою часто є оптимізація функції якості кластеризації, яка винагороджує компактні кластери та добре розділені різні кластери. На практиці існує багато алгоритмів кластеризації, що відбивають різні визначення «кластера»: деякі методи шукають компактні сферичні кластери; інші визначають кластери як суміжні області високої густини; ще інші знаходять гнучкі неопуклі групування на основі зв'язності у графі чи статистичній моделі. Нижче викладено математичні основи чотирьох важливих технік кластеризації: спектральної кластеризації, агломеративної ієрархічної кластеризації, k-means та гаусових змішаних моделей (GMM).

3.1. Спектаральна кластеризація

Спектральна кластеризація здійснює групування через власну структуру графа схожості. Будуємо неорієнтований зважений граф $G = (V, E)$, де кожна вершина $i \in V$ відповідає точці даних $\mathbf{x}i$, а ваги ребер $w_{ij} \geq 0$ кодують схожість між точками i та j. Граф описується зваженою матрицею суміжності $W = [w_{ij}]$ і матрицею ступенів $D = \text{diag}(d_1, \ldots, d_n)$, де $d_i = \sum_{j=1}^{n} w_{ij}$. Лапласіан графа задається як

$$L = D - W, \tag{3.1}$$

і є симетричною додатно напіввизначеною матрицею. Можна також використовувати нормалізований Лапласіан, наприклад симетричну нормалізовану форму

$L_{\text{sym}} = I - D^{-1/2} W D^{-1/2}$ або форму випадкового блукання $L_{\text{rw}} = I - D^{-1} W$. Ключове спостереження: малі власні значення та власні вектори L виявляють структуру кластерів. Зокрема, якщо граф має K зв'язних компонент, L матиме рівно K нульових власних значень з ортогональними власними векторами-індикаторами цих компонент. Загальніше, добре розділені кластери відповідають власним векторам з малими власними значеннями, які апроксимують такі індикатори.

Отже, кластеризацію можна сформулювати як задачу розбиття графа. Наприклад, однією з цілей є нормалізований розріз:

$$\text{Ncut}(C_1, \ldots, C_K) = \sum_{k=1}^{K} \frac{\text{cut}(C_k, \bar{C}_k)}{\text{vol}(C_k)} \,, \tag{3.2}$$

де $\text{cut}(A, B) = \sum_{i \in A, , j \in B} w_{ij}$ — загальна вага ребер між множиною A та її доповненням, а $\text{vol}(A) = \sum_{i \in A} d_i$ — обсяг (загальний ступінь) кластера A. Мінімізувати (3.2) для всіх K-шляхових розбиттів NP-важко, але спектральна кластеризація знаходить наближене розв'язання, розслаблюючи задачу до неперервних змінних. У розслабленій постановці індикаторним вектором дозволяють набувати дійсних значень, що веде до оптимізації, розв'язуваної через власний розклад L. Зокрема, обчислюють K власних векторів $v_1, \ldots, v_K$, що відповідають K найменшим власним значенням обраного Лапласіана. Ці вектори формують матрицю вкладення $V = [v_1; v_2; \cdots; v_K] \in \mathbb{R}^{n \times K}$. Кожен рядок i матриці V задає K-вимірне відображення точки $\mathbf{x}_i$ у спектральному просторі. Нарешті, на цих вкладених точках застосовують просту кластеризацію (наприклад, k-means), щоб отримати K кластерів. Процедура ефективно знаходить групи точок, що сильно зв'язані у графі схожості, навіть якщо вони не є опуклими в початковому просторі ознак. Метод ґрунтується на лінійній алгебрі та теорії графів: він оптимізує розслаблений розріз графа, а якість розв'язку пов'язана з фундаментальними результатами спектральної теорії (наприклад, нерівністю Чігера). Відзначимо, що спектральна кластеризація (зокрема варіант normalized cut за Ши та Маліком) успішно застосовується до сегментації зображень, розглядаючи пікселі як вершини графа та афінності — на основі схожості пікселів. Теоретична перевага методу — здатність вловлювати неопуклі структури та гарантія глобального оптимуму для розслабленої задачі (оскільки обчислення власних векторів є опуклим), хоча остаточне дискретне розбиття є лише наближенням цього оптимуму.

3.2. Агломеративна кластеризація

Агломеративна ієрархічна кластеризація будує багаторівневу ієрархію кластерів, поступово об'єднуючи менші кластери у більші. Вона починає з найдрібнішого розбиття, де кожна точка — окремий кластер, ітеративно зливаючи дві найсхожіші групи, доки не залишиться один кластер (або доки не буде досягнуто бажаного числа кластерів). Формально, припустимо, що на певному кроці маємо кластеризацію $\mathcal{C} = \{C_1, \ldots, C_m\}$. Визначимо функцію зв'язку (міру міжкластерної відстані) $\Delta(C_i, C_j)$ для будь-якої пари кластерів. Поширені варіанти: одноланцюговий зв'язок $\Delta_{\min}(A, B) = \min_{x \in A,, y \in B} \|x - y\|$ та повний зв'язок $\Delta_{\max}(A, B) = \max_{x \in A,, y \in B} \|x - y\|$ та ін. Алгоритм знаходить пару (C_p, C_q) з мінімальною відстанню $\Delta(C_p, C_q)$ і зливає їх: $C_{pq} = C_p \cup C_q$. Процес жадібний; здійснене злиття не скасовується, а послідовність злиттів визначає бінарне дерево кластерів — дендрограму. Дендрограма кодує кластеризації всіх масштабів — від n одиничних кластерів до одного кластера — шляхом «обрізання» дерева на різних рівнях.

Перевага агломеративної кластеризації полягає у відсутності потреби в метричному просторі — достатньо матриці попарних відстаней. Критерій злиття (зв'язок) визначає форму та властивості кінцевих кластерів: одноланцюговий зв'язок схильний утворювати витягнуті «ланцюгові» кластери, тоді як повний зв'язок дає компактні, щільно зв'язані групи, а середній зв'язок забезпечує компроміс. Теоретично дендрограма задає ультраметрику, що апроксимує початкову відстань. Якщо дані мають ієрархічну природу, правильний вибір зв'язку може відтворити цю структуру. Однак жадібний характер означає, що метод не гарантує глобального оптимуму для загальної функції якості (окрім самого критерію зв'язку). Попри це, агломеративна кластеризація популярна завдяки простоті та інтерпретованості дерева кластерів. Метод детермінований за фіксованої матриці відстаней і правила зв'язку та завершується після $n - 1$ злиттів; наївна реалізація має складність $O(n^2 \log n)$, яку можна покращити за допомогою ефективних структур даних.

3.3. Метод К-середніх

Алгоритм k-means — класичний метод частинної кластеризації, який безпосередньо мінімізує внутрішньокластерну дисперсію. Для заданого числа кластерів K k-means шукає розбиття $\{C_1, \ldots, C_K\}$ даних і центроїди $\boldsymbol{\mu}1, \ldots, \boldsymbol{\mu}K \in \mathbb{R}^d$, що мінімізують суму квадратів відстаней від кожної точки до центроїда її класте-

ра:

$$\min_{C_1,\dots,C_K} J = \sum_{k=1}^{K} \sum_{i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 , \tag{3.3}$$

де $\boldsymbol{\mu}k = \frac{1}{|C_k|} \sum_{i \in C_k} \mathbf{x}_i$ — середнє кластера C_k. Оптимізація (3.3) є неопуклою та NP-складною для загальних K і d. Проте відомий алгоритм Ллойда — ефективний евристичний метод, що, як правило, знаходить хороший локальний мінімум. Алгоритм циклічно виконує два кроки: (i) присвоєння кожної точки найближчому центроїду та (ii) оновлення центроїдів як середніх їхніх точок. Кожна ітерація не збільшує J, тому збіжність до локального мінімуму гарантовано. На практиці k-means швидко сходиться, хоча результат залежить від ініціалізації.

З теоретичної точки зору k-means можна розглядати як граничний випадок кластеризації GMM за припущення рівних сферичних коваріацій і жорстких належностей. Він розбиває простір даних на K осередків Вороного; межі кластерів — гіперплощини, рівновіддалені між центроїдами. Тому k-means найефективніший, коли істинні кластери приблизно сферичні або опуклі та добре розділені. Простість k-means — перевага: одна ітерація має складність $O(nKd)$, тож метод часто є першим вибором для великих даних.

3.4. Гаусова змішана модель

Гаусова змішана модель (GMM) підходить до кластеризації імовірнісно, моделюючи розподіл даних як суміш K гаусових компонент. У GMM кожен кластер відповідає компоненті — d-вимірному нормальному розподілу з власними середнім $\boldsymbol{\mu}k$ та коваріаційною матрицею Σ_k. Нехай π_k — вага компоненти k, $\pi_k \geq 0$, $\sum_{k=1}^{K} \pi_k = 1$. Ймовірнісна густина моделі:

$$p(\mathbf{x}; \{\pi_k, \boldsymbol{\mu}k, \Sigma_k\}_{k=1}^{K}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}k, \Sigma_k), \tag{3.4}$$

де $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}k, \Sigma_k)$ — гаусова густина компоненти k. GMM задає м'яку кластеризацію: кожна точка $\mathbf{x}_i$ належить кожному кластеру k з певною ймовірністю (відповідальністю)

$$\gamma ik = P(z_i = k \mid \mathbf{x}_i) = \frac{\pi_k \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j)}, \tag{3.5}$$

де z_i — латентна мітка кластера для $\mathbf{x}_i$. Параметри моделі оцінюють максимальною правдоподібністю; лог-правдоподібність даних $X = \{\mathbf{x}1, \ldots, \mathbf{x}n\}$:

$$\mathcal{L}(\pi, \mu, \Sigma) = \sum_{i=1}^{n} \ln\Big(\sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}k, \Sigma_k)\Big). \tag{3.6}$$

Оскільки (3.6) не є опуклою, застосовують алгоритм EM, який чергує крок очікування (E) — обчислення γ_{ik} за (3.5), та крок максимізації (M) — оновлення параметрів

$$\boldsymbol{\mu}k^{\text{new}} = \frac{\sum_{i=1}^{n} \gamma_{ik} \, \mathbf{x}_i}{\sum_{i=1}^{n} \gamma_{ik}}, \tag{3.7}$$

$$\Sigma_k^{\text{new}} = \frac{\sum_{i=1}^{n} \gamma_{ik} \, (\mathbf{x}_i - \boldsymbol{\mu}k^{\text{new}})(\mathbf{x}_i - \boldsymbol{\mu}k^{\text{new}})^{\top}}{\sum_{i=1}^{n} \gamma_{ik}}, \tag{3.8}$$

$$\pi_k^{\text{new}} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ik}. \tag{3.9}$$

Кожна ітерація EM не зменшує (3.6), тому алгоритм монотонно покращує правдоподібність і сходиться до стаціонарної точки. Підсумкові параметри $\{\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}k\}$ задають K гаусових кластерів. Перевага GMM — гнучкість: дозволивши повні коваріаційні матриці, кластери можуть мати довільні еліпсоїдні форми. Для великих K суміші гаусів апроксимують будь-яку неперервну густину з довільною точністю. Імовірнісна рамка дає змогу робити обґрунтований вибір K (наприклад, за BIC) і кількісно оцінювати невизначеність належності точок через γ_{ik}. Хоча правдоподібність може мати багато локальних максимумів і EM чутливий до ініціалізації, GMM забезпечує надійну статистичну основу для кластеризації, поєднуючи її з генеративним моделюванням: кожен кластер є прихованою компонентою генеративної моделі, а навчання кластерів відповідає оцінюванню параметрів багатомодального розподілу.

4 ЧУТЛИВІСТЬ ТА СПЕЦИФІЧНІСТЬ

Під час оцінювання кластеризації (за наявності еталонних класових міток) показники чутливості та специфічності, запозичені з бінарної класифікації, кількісно характеризують здатність групування відокремлювати певний клас від інших. Кожен показник відбиває окремий тип помилок кластеризації: чутливість відображає, яку частку елементів класу вдалося згрупувати разом (мінімізація хибнонегативних результатів), тоді як специфічність показує, наскільки добре алгоритм уникнув домішування елементів інших класів (мінімізація хибнопозитивних результатів). Нижче наведено формальні визначення обох показників і їхню інтерпретацію на прикладі відокремлення групи захворювання від контрольної групи.

4.1. Чутливість

Чутливість (true positive rate) вимірює частку справжніх позитивних випадків, які правильно ідентифіковані кластеризацією. Формально, якщо певний кластер відповідає цільовому класу, чутливість дорівнює частці елементів цього класу, що потрапили у відповідний кластер. Нехай TP — кількість істинних позитивних (об'єктів цільового класу, розміщених у кластері класу), а FN — кількість хибнонегативних (об'єктів того самого класу, які алгоритм не включив до цільового кластера). Тоді

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \tag{4.1}$$

тобто частка членів класу, захоплених кластером. Висока чутливість (наближена до 1) означає, що майже всі справжні елементи класу згруповані разом, тобто алгоритм пропустив дуже мало об'єктів (мало FN). Натомість низька чутливість указує, що значна частина справжніх членів класу розподілилася по інших кластерах, і кластеризація не змогла належно виявити цей клас.

4.2. Специфічність

Специфічність (true negative rate) вимірює частку справжніх негативних випадків, які правильно виключено з кластера цільового класу. Вона оцінює «чи-

стоту» кластера щодо присутності сторонніх елементів. Нехай TN — кількість істинних негативних (об'єктів нецільового класу, що правильно не потрапили до кластера), а FP — кількість хибнопозитивних (об'єктів інших класів, помилково включених у кластер цільового класу). Тоді

$$\text{Specificity} = \frac{TN}{TN + FP}, \tag{4.2}$$

тобто частка «чужих» об'єктів, успішно виключених із кластера. Висока специфічність (близька до 1) означає, що кластер майже не містить елементів інших класів (мінімум FP). Низька специфічність свідчить, що кластер «контамінований» значною кількістю сторонніх елементів, і класи недостатньо розділені.

Отже, висока чутливість гарантує, що кластер охоплює практично всі елементи свого класу, тоді як висока специфічність гарантує, що у кластері майже немає представників інших класів. Ідеальна кластеризація, спрямована на розділення класів повинна досягати і високої чутливості, і високої специфічності, забезпечуючи повне відтворення кожного класу без перехресного змішування. Разом ці показники дають всебічне уявлення про якість кластеризації: чутливість фокусує увагу на уникненні пропусків елементів класу, а специфічність — на запобіганні помилковому включенню елементів інших класів.

5 РЕЗУЛЬТАТИ

У цьому розділі подано двокластерні розв'язання, отримані чотирма неконтрольованими алгоритмами: Spectral Clustering, Agglomerative Clustering (середній зв'язок), k-Means та двокомпонентна гаусова суміш (GMM).[1] Проведено два взаємодоповнювальних експерименти:

— Загальна схожість. Одну матрицю схожості отримували усередненням по всіх комбінаціях фільтр–канал (blue, green, red та grey).

— Кольороспецифічна схожість. Окремі матриці схожості обчислювали для каналів blue, green і red (усереднені за фільтрами).

Ознайомитись з програмним кодом можна в Додатку А; нижче зосередимось на чутливості (TPR) та специфічності (TNR), подаючи контингенційні підрахунки лише текстом.

5.1. Загальна схожість

Спектральна кластеризація забезпечує помірне відновлення ядер ВС (TPR $=$ 0.529) і водночас найчистіший контрольний кластер (TNR $=$ 0.701).Вузьке ядро (σ $=$ 0.1) обмежує афінність локальним околами, віддаючи перевагу *чистому* контрольному кластеру ціною фрагментації розподілу ВС.

Агломеративна кластеризація підвищує чутливість до 0.564, але вводить більше контрольних ядер у кластер ВС (TNR $=$ 0.644). Середній зв'язок зливає межеві об'єкти, коли міжкласові відстані наближаються до внутрішньокласових, що й зумовлює змішане членство.

k-Means пропонує проміжний компроміс (TPR $=$ $0.549,$ TNR $=$ 0.690). Його лінійне розбиття Вороного не фіксує неопуклу геометрію простору схожості, але приблизно поділяє дві модальні області.

Гаусова суміш. Дозволивши повні коваріації, отримуємо найкраще відновлення ВС (TPR $=$ 0.588) за найменшої специфічності (TNR $=$ 0.632): алгоритм ЕМ відносить неоднозначні контрольні точки до щільнішої (ВС-домінантної) компоненти, збільшуючи кількість хибнопозитивних.

[1] Усі кластери маркували *постфактум*, тобто кластер, що максимізував Sensitivity + Specificity, оголошували позитивним (ВС).

5.2. Кольороспецифічна схожість

5.2.1. Синій канал

Подібність у Blue підкреслює поглинання хроматину; у результаті спектральна кластеризація виділяє компактне ядро ВС (TPR $= 0.735$) з *досконалою* чистотою (TNR $= 1.000$). Середній зв'язок майже всю вибірку зливає в один метакластер, що дає ідеальну чутливість, але практично нульову специфічність (0.034). І k-Means, і GMM сходяться до тотожного сферичного розбиття (TPR $= 0.544$, TNR $= 0.586$), що означає майже ізотропний розподіл після нормалізації контрасту.

5.2.2. Зелений канал

Зелений канал надає найсильніший сигнал: спектральна кластеризація досягає найвищої сумарної ефективності (TPR $= 0.868$, TNR $= 0.966$). Межа графа практично збігається з повним розділенням класів, лишаючи лише кілька ядер ВС в контрольному кластері. Середній зв'язок знову переоб'єднує, а обидві модельні техніки відтворюють синій результат, що свідчить: після нормалізації гістограм канальні відмінності для цих методів згладжуються.

5.2.3. Червоний канал

Інтенсивності у Red займають проміжну позицію. Спектральний метод дає TPR $= 0.765$ при бездоганній специфічності; k-Means і GMM відновлюють близько 59 % ядер ВС з TNR ≈ 0.56. Артефакт середнього зв'язку повторюється, підтверджуючи, що ефект «ланцюжка» притаманний самому критерію, а не шуму каналу.

5.3. Інтерпретація

Виділено дві тенденції:

— Чутливість каналів. Хроматин, забарвлений реакцією Фельґена, максимально контрастує у зеленій смузі, помірно — у червоній і найменше — у синій. Тому графова кластеризація (що використовує локальну афінність) показує найкращі результати на Green і найгірші — на усередненій матриці, де сигнал Green розбавляється менш інформативними каналами.

— Упередженість зв’язку. Середній зв’язок систематично максимізує чутливість, створюючи гігантський кластер, однак ціною катастрофічної втрати специфічності. Ефект незалежний від каналу, що вказує на «ланцюгову» природу самого критерію.

Наслідки. Перспективною видається канал-орієнтована ансамблева стратегія: залишити *Green–spectral* розбиття як високоточне ядро, а далі поступово розширювати його за допомогою ймовірнісного (GMM) приєднання, щоб «врятувати» решту ядер BC. Такий гібридний підхід має підвищити як чутливість, так і специфічність понад рівні, досягнуті окремими моделями.

6 ВИСНОВКИ

Курсова робота була присвячена дослідженню фрактальної структури контурів ядер букального епітелію з метою виявлення статистичних відмінностей між пацієнтками з раком молочної залози (група *BC*) та здоровим контролем (*Control*). Нижче викладено узагальнені підсумки виконаних етапів, отримані результати та можливі напрями подальшого застосування.

6.1. Основні етапи роботи

— Формування вибірки. Опрацьовано 322 цифрові сканограми ядер ($n_{\text{BC}} = 204$, $n_{\text{Ctrl}} = 118$), отримані в трьох спектральних каналах (Blue, Green, Red) та кількох варіантах освітлення.

— Оцінка схожості.Запроваджено метрику p-статистики $\mu(x, y)$ як частку інтервалів порядкових статистик, для яких емпірична частота узгоджується з теоретичним розподілом при 95 %-вому довірчому рівні (межі Вілсона, параметр $g = 3$). Обчислено повні матриці схожості як загалом, так і для кожного каналу окремо.

— Кластеризація. Досліджено чотири некеровані алгоритми (Spectral, Agglomerative k-Means, GMM) у постановці двокласового поділу. Для кожного розбиття визначено чутливість (TPR) та специфічність (TNR) відносно еталонних лікарських міток.

6.2. Ключові результати

— Компактність розподілу. Розрахована p-статистика демонструє, що вибірки ядер групи *BC* утворюють відносно однорідний кластер у просторі фрактальних розмірностей; водночас контрольні зразки розташовані більш розріджено та частково перекриваються з *BC*.

— Найкраще відокремлення каналів. У зеленому (Green) каналі спектральна кластеризація досягає показників $\text{TPR} = 0.868$ та $\text{TNR} = 0.966$, що свідчить про високий міжкласовий контраст саме на цій довжині хвилі. Канали Blue та Red також дають прийнятні результати, але з нижчою чутливістю.

— Баланс TPR/TNR. Серед усередненої (загальної) матриці схожості найкращий компроміс спостерігається для GMM ($\text{TPR} = 0.588,\ \text{TNR} = 0.632$), що

свідчить про потенційну неоднорідність підгруп BC у повному спектральному просторі.

6.3. Узагальнені висновки

— Доведено статистичну компактність фрактальних розмірностей ядер букального епітелію у пацієнток із раком молочної залози порівняно з контрольною групою. Це підтверджує гіпотезу, що злоякісні клітини мають більш стійку та повторювану текстурну структуру ядра.

— Найінформативнішим виявився зелений канал. Висока оптична контрастність сигналу Feulgen у Green-діапазоні забезпечила найкраще розділення класів за спектральною кластеризацією.

— Показано придатність p-статистики як універсальної симетричної міри, що не потребує параметричних припущень і може застосовуватися у різних каналах та фільтрах.

6.4. Перспектива практичного використання

Отримані дані створюють підґрунтя для розроблення моделі штучного інтелекту, яка автоматично класифікуватиме сканограми ядер та, відповідно, виконуватиме раннє неінвазивне виявлення раку молочної залози.

Планується:

— використати зелений спектральний канал як основне вхідне представлення;

— інтегрувати p-статистику у вигляді ядра схожості для напівавідірваних або глибоких кластеризаторів;

— розширити вибірку та провести валідацію на незалежних центрах.

Завдяки низькій вартості букального мазка та відсутності інвазивних процедур запропонована методика може стати додатковим скринінговим інструментом клінічної практики, збільшуючи шанси на своєчасну діагностику захворювання.

Література

[1] Aloise D., Deshpande A., Hansen P., Popat P., *NP-hardness of Euclidean sum-of-squares clustering*, Machine Learning, **75** (2009), 245–249.

[2] Andrushkiw R. I., Klyushin D. A., Petunin Yu. I., Savkina M. Yu., *Exact confidence limits for unknown probability in Bernoulli models*, Proc. 27th Int. Conf. on Information Technology Interfaces (ITI 2005), 164–168.

[3] Bishop C. M., *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[4] Chung F. R. K., *Spectral Graph Theory*. Providence, RI: American Mathematical Society, 1997.

[5] Dempster A. P., Laird N. M., Rubin D. B., *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, Series B, **39** (1977), 1–38.

[6] Derrick B., White P., Toher D., *Parametric and non-parametric tests for the comparison of two samples which both include paired and unpaired observations*, Journal of Modern Applied Statistical Methods, **18** (2019), eP2854.

[7] Jain A. K., Murty M. N., Flynn P. J., Data clustering: A review, *ACM Computing Surveys*, **31** (1999), 264–323.

[8] Kaufman L., Rousseeuw P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley, 1990.

[9] Klyushin D. A., Petunin Yu. I., *A nonparametric test for the equivalence of populations based on a measure of proximity of samples*, Ukrainian Mathematical Journal, **55** (2003), 181–198.

[10] Lloyd S., Least squares quantization in PCM, *IEEE Transactions on Information Theory*, **28** (1982), 129–137.

[11] von Luxburg U., A tutorial on spectral clustering, *Statistics and Computing*, **17** (2007), 395–416.

[12] Matveichuk S. A., Petunin Yu. I., *Generalization of Bernoulli schemes that arise in order statistics. II*, Ukrainian Mathematical Journal, **43** (1991), 728–734.

[13] Madreimov I., Petunin Yu. I., *Characterization of a uniform distribution using order statistics*, Theory of Probability and Mathematical Statistics, **27** (1982), 96–102.

[14] Shi J., Malik J., Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (2000), 888–905.

[15] Vysochanskii D., Petunin Yu. I., *Justification of the* 3σ *rule for unimodal distributions*, Theory of Probability and Mathematical Statistics, **21** (1980), 25–36.

Додаток А Повні лістинги програмного коду

Усі скрипти написані мовою Python 3.12 та розроблялися у середовищі PyCharm Professional 2024.1. Використані бібліотеки: NumPy 1.26, SciPy 1.13, scikit - learn 1.4, Matplotlib 3.9, Pillow 10.3. Нижче наведено вихідні коди трьох файлів у порядку їх імпортів.

A.1 compute_similarity.py

Listing 1: Файл compute_similarity.py

```
from pathlib import Path
import numpy as np
import pickle


def read_series(path: Path) -> np.ndarray:
    return np.loadtxt(path, dtype=float)


def load_data(data_root: Path) -> dict[str, dict[str, dict[str, np.
   ↪ ndarray]]]:
    data: dict[str, dict[str, dict[str, np.ndarray]]] = {"BC": {}, "
       ↪ Control": {}}
    for group in data:
        folder = data_root / group
        if not folder.exists():
            raise FileNotFoundError(f"Не знайденопапку  {folder}")
        for filepath in folder.glob("*_*"):
            name, channel = filepath.stem.split("_", 1)
            series = read_series(filepath)
            data[group].setdefault(name, {})[channel] = series
    return data


def compute_p_statistic(x_sorted, y, g=3):
    n = x_sorted.shape[0]
    m = y.shape[0]

    i_idx, j_idx = np.triu_indices(n, k=1)
```

```
    p0 = (j_idx - i_idx) / (n + 1)

    counts = [((x_sorted[i] <= y) & (y <= x_sorted[j])).sum()
              for i, j in zip(i_idx, j_idx)]
    h = np.array(counts, float) / m
    neff = m + g * g
    a = h * m + 0.5 * g * g
    b = g * np.sqrt(h * (1 - h) * m + 0.25 * g * g)
    p_low = (a - b) / neff
    p_up = (a + b) / neff

    I = (p_low <= p0) & (p0 <= p_up)

    return I.mean()


if __name__ == "__main__":
    project_root = Path("/Users/nekomelisa/Documents/jez/code/data").
       ↪ parent
    data_root = project_root / "data"

    data = load_data(data_root)

    for group in data:
        for sample_id, channels in data[group].items():
            for channel, arr in channels.items():
                arr.sort()

    bc_list, bc_labels = [], []
    for pid, channels in data["BC"].items():
        for ch, arr in channels.items():
            bc_list.append(arr)
            bc_labels.append(f"{pid}_{ch}")

    ctrl_list, ctrl_labels = [], []
    for pid, channels in data["Control"].items():
        for ch, arr in channels.items():
            ctrl_list.append(arr)
            ctrl_labels.append(f"{pid}_{ch}")

    n_bc = len(bc_list)
```

```
    S_bc = np.zeros((n_bc, n_bc), dtype=float)
    for i in range(n_bc):
        for j in range(n_bc):
            S_bc[i, j] = compute_p_statistic(bc_list[i], bc_list[j],
               ↪ g=1.96)

    m_ctrl = len(ctrl_list)
    S_ctrl = np.zeros((m_ctrl, m_ctrl), dtype=float)
    for i in range(m_ctrl):
        for j in range(m_ctrl):
            S_ctrl[i, j] = compute_p_statistic(ctrl_list[i],
               ↪ ctrl_list[j], g=1.96)

    S_cross = np.zeros((n_bc, m_ctrl), dtype=float)
    for i in range(n_bc):
        for j in range(m_ctrl):
            S_cross[i, j] = compute_p_statistic(bc_list[i], ctrl_list
               ↪ [j], g=1.96)

    payload = {
        "S_bc": S_bc,
        "S_ctrl": S_ctrl,
        "S_cross": S_cross,
        "bc_labels": bc_labels,
        "ctrl_labels": ctrl_labels
    }
    with open("sim_matrices.pkl", "wb") as f:
        pickle.dump(payload, f)
```

A.2 cluster_analysis.py

Listing 2: Файл cluster_analysis.py

```
import numpy as np
import pickle
import matplotlib.pyplot as plt

from sklearn.cluster import (
    SpectralClustering,
    KMeans,
    AgglomerativeClustering,
```

```
)
from sklearn.mixture import GaussianMixture
from sklearn.manifold import MDS
from collections import Counter, defaultdict


def print_and_plot(labels, group_tags, method_name):
    counts = defaultdict(Counter)
    for lbl, tag in zip(labels, group_tags):
        counts[lbl][tag] += 1

    print(f'\n=== {method_name} ===')
    for cid in sorted(counts):
        bc_c = counts[cid]['BC']
        ctrl_c = counts[cid]['Control']
        print(f'Cluster {cid}:  BC={bc_c:3d},  Control={ctrl_c:3d}')

    clusters = sorted(counts)
    bc_counts = [counts[c]['BC'] for c in clusters]
    ctrl_counts = [counts[c]['Control'] for c in clusters]

    x = np.arange(len(clusters))
    width = 0.4

    plt.figure(figsize=(6, 4))
    plt.bar(x - width / 2, bc_counts, width, label='BC')
    plt.bar(x + width / 2, ctrl_counts, width, label='Control')
    plt.xticks(x, clusters)
    plt.xlabel('Cluster ID')
    plt.ylabel('Count')
    plt.title(method_name)
    plt.legend()
    plt.tight_layout()
    plt.show()


if __name__ == '__main__':
    with open('sim_matrices_3.pkl', 'rb') as f:   # для g = 3.0
        data = pickle.load(f)

    S_bc = data['S_bc']
```

```
S_ctrl = data['S_ctrl']
S_cross = data['S_cross']
bc_labels = data['bc_labels']
ctrl_labels = data['ctrl_labels']

top = np.hstack([S_bc, S_cross])
bottom = np.hstack([S_cross.T, S_ctrl])
S_all = np.vstack([top, bottom])
d_all = 1.0 - S_all
d_all = (d_all + d_all.T) / 2

sigma = 0.1
affinity = np.exp(-(d_all ** 2) / (2 * sigma ** 2))

group_tags = ['BC'] * len(bc_labels) + ['Control'] * len(
    ↪ ctrl_labels)

mds = MDS(
    n_components=2,
    dissimilarity='precomputed',
    random_state=0
)
X_mds = mds.fit_transform(d_all)

clustering_methods = {
    'SpectralClustering σ(=0.1)': SpectralClustering(
        n_clusters=2,
        affinity='precomputed',
        assign_labels='kmeans',
        random_state=0
    ),
    'AgglomerativeClustering (avg linkage)':
      ↪ AgglomerativeClustering(
        n_clusters=2,
        metric='precomputed',
        linkage='average'
    ),
    'KMeans (k=2)': KMeans(
        n_clusters=2,
        random_state=0
    ),
```

```
        'GaussianMixture (n=2)': GaussianMixture(
            n_components=2,
            covariance_type='full',
            random_state=0
        )
    }

    for name, algo in clustering_methods.items():
        if 'SpectralClustering' in name:
            labels = algo.fit_predict(affinity)
        elif name.startswith(('DBSCAN', 'OPTICS', '
           ↪ AgglomerativeClustering')):
            labels = algo.fit_predict(d_all)
        else:
            labels = algo.fit_predict(X_mds)

        print_and_plot(labels, group_tags, name)
```

A.3 colour_claster.py

Listing 3: Файл colour_claster.py

```
from pathlib import Path
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages

from sklearn.cluster import (
    SpectralClustering,
    KMeans,
    AgglomerativeClustering,
)
from sklearn.mixture import GaussianMixture
from sklearn.manifold import MDS
from collections import Counter, defaultdict

from compute_similarity import load_data, compute_p_statistic

sigma = 0.1
g = 3
```

```
clustering_methods = {
    'SpectralClustering': SpectralClustering(
        n_clusters=2,
        affinity='precomputed',
        assign_labels='kmeans',
        random_state=0
    ),
    'AgglomerativeClustering': AgglomerativeClustering(
        n_clusters=2,
        metric='precomputed',
        linkage='average'
    ),
    'KMeans': KMeans(n_clusters=2, random_state=0),
    'GaussianMixture': GaussianMixture(n_components=2,
      ↪ covariance_type='full', random_state=0)
}

if __name__ == '__main__':
    BASE_DIR = Path(__file__).resolve().parents[1]
    DATA_ROOT = BASE_DIR / "data"
    print(f"Using data directory: {DATA_ROOT}")

    data = load_data(DATA_ROOT)
    channels = sorted({ch for grp in data.values() for ch in next(
      ↪ iter(grp.values())).keys()})

    output_pdf = BASE_DIR / 'all_clusters.pdf'
    with PdfPages(output_pdf) as pdf:
        for ch in channels:
            bc_items = [(pid, data['BC'][pid][ch]) for pid in data['
              ↪ BC'] if ch in data['BC'][pid]]
            ctrl_items = [(pid, data['Control'][pid][ch]) for pid in
              ↪ data['Control'] if ch in data['Control'][pid]]

            bc_list = [arr for pid, arr in bc_items]
            ctrl_list = [arr for pid, arr in ctrl_items]
            n_bc, m_ctrl = len(bc_list), len(ctrl_list)
            group_tags = ['BC'] * n_bc + ['Control'] * m_ctrl

            S_bc = np.zeros((n_bc, n_bc), float)
            S_ctrl = np.zeros((m_ctrl, m_ctrl), float)
```

```
S_cross = np.zeros((n_bc, m_ctrl), float)
for i in range(n_bc):
    for j in range(n_bc):
        S_bc[i, j] = compute_p_statistic(bc_list[i],
            ↪ bc_list[j], g=g)
for i in range(m_ctrl):
    for j in range(m_ctrl):
        S_ctrl[i, j] = compute_p_statistic(ctrl_list[i],
            ↪ ctrl_list[j], g=g)
for i in range(n_bc):
    for j in range(m_ctrl):
        S_cross[i, j] = compute_p_statistic(bc_list[i],
            ↪ ctrl_list[j], g=g)

top = np.hstack([S_bc, S_cross])
bottom = np.hstack([S_cross.T, S_ctrl])
S_all = np.vstack([top, bottom])
S_all = (S_all + S_all.T) / 2
d_all = 1.0 - S_all
affinity_all = np.exp(-(d_all ** 2) / (2 * sigma ** 2))

mds_model = MDS(n_components=2, dissimilarity='
   ↪ precomputed', random_state=0)
X_mds = mds_model.fit_transform(d_all)

print(f"\n===== CHANNEL: {ch} =====")
for name, algo in clustering_methods.items():
    if name == 'SpectralClustering':
        labels = algo.fit_predict(affinity_all)
    elif name in ('AgglomerativeClustering',):
        labels = algo.fit_predict(d_all)
    else:
        labels = algo.fit_predict(X_mds)

    counts = defaultdict(Counter)
    for lbl, tag in zip(labels, group_tags):
        counts[lbl][tag] += 1
    print(f"\n{name} — channel={ch}")
    for cid in sorted(counts):
        print(f" Cluster {cid}: BC={counts[cid]['BC']:3d
            ↪ }, Control={counts[cid]['Control']:3d}")
```

```
            if len(counts) == 2:
                cids = sorted(counts)
                pos_c = max(cids, key=lambda c: counts[c]['BC'])
                neg_c = min(cids) if max(cids) == pos_c else [c
                   ↪ for c in cids if c != pos_c][0]

                TP = counts[pos_c]['BC']
                FN = counts[neg_c]['BC']
                TN = counts[neg_c]['Control']
                FP = counts[pos_c]['Control']
                sensitivity = TP / (TP + FN) if (TP + FN) > 0
                   ↪ else float('nan')
                specificity = TN / (TN + FP) if (TN + FP) > 0
                   ↪ else float('nan')
                print(f" Sensitivity: {sensitivity:.3f},
                   ↪ Specificity: {specificity:.3f}")


            bc_counts = [counts[c]['BC'] for c in sorted(counts)]
            ctrl_counts = [counts[c]['Control'] for c in sorted(
               ↪ counts)]
            x = np.arange(len(bc_counts))
            fig, ax = plt.subplots(figsize=(6, 4))
            ax.bar(x - 0.2, bc_counts, 0.4, label='BC')
            ax.bar(x + 0.2, ctrl_counts, 0.4, label='Control')
            ax.set_xticks(x)
            ax.set_xticklabels(sorted(counts))
            ax.set_xlabel('Cluster ID')
            ax.set_ylabel('Count')
            ax.set_title(f"{name} — channel={ch}")
            ax.legend()
            plt.tight_layout()

            pdf.savefig(fig)
            plt.close(fig)
```