

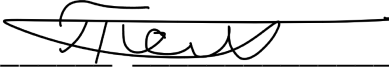
## Отчет по преддипломной практике

Подготовка выпускной квалификационной работы бакалавра  
на тему:

*Разработка метода определения экспрессии генов по  
изображениям мозга плодовой мушки*

по направлению 01.03.02 «Прикладная математика и  
информатика»

(профиль «Биоинформатика»)

Студент группы 3630102/70401 \_\_\_\_\_  Р.А. Темиргалиев  
Оценка научного руководителя ВКР: \_\_\_\_\_ К.Н. Козлов

## СОДЕРЖАНИЕ

Введение .....	2
Глава 1. Постановка задачи и описание исходных данных.....	3
1.1. Описание исходных данных .....	3
1.2. Сложность задачи .....	4
1.3. Существующие методы решений .....	5
Глава 2. Описание используемых методов .....	7
2.1. Методы обработки изображений реализованные в пакете ProStack.....	7
2.2. Инструмент поиска автофлуоресценции. AFid .....	13
2.2.1. Входные данные и требования.....	13
2.2.2. Создание маски пересечения.....	13
2.2.3. Кластеризация для идентификации автофлуоресценции .....	14
2.2.4. Расширение автофлуоресцентных областей.....	15
2.2.5. Обзор алгоритма .....	17
Глава 3. Применение описанных методов .....	18
3.1. Модификация AFid.....	18
3.1.1. Переписывание на Python .....	18
3.1.2. Предобработка входных данных .....	18
3.2. Применение AFid .....	20
3.3. Проверка статистической гипотезы .....	20
3.4. Кластеризация мозга мушки .....	20
3.5. Название параграфа .....	21
Глава 4. Результаты исследования и сравнительный анализ.....	22
4.1. Результат проверки статистической гипотезы .....	22
4.2. Результаты кластеризации и фильтрации.....	22
4.2.1. Результат фильтрации.....	22
Глава 5. Заключение .....	25
Глава 6. Выводы .....	26
Список использованных источников.....	27

## ВВЕДЕНИЕ

Для исследования генной регуляции требуется получать количественные данные по экспрессии генов с учетом пространственной локализации.

Рассмотрим следующую задачу для решения которой изучается генная регуляция. Итак, сначала по изображениям мозга плодовой мушки измеряют уровень экспрессии генов, то есть получают количественные данные с учетом пространственной локализации генов. Далее полученные данные используют для изучения поведения мух в период спаривания - сравнивают уровни экспрессии генов в разных частях мозга у мушек разных полов с их поведением в период спаривания. Под поведением можно понимать их привлекательность друг другу, желание спариваться и др. Далее эти статистические связи от модельных объектов(мушек) можно попробовать распространить на более сложные организмы(мыши, собаки и др.)

В данной работе изучается получение количественных данных которые можно было бы использовать для приведенной задачи выше. Для выделения на экспериментальных изображениях комплексов молекул РНК будут использованы методы обработки изображений.

Целью данной работы является разработка алгоритма для выделения на экспериментальных изображениях комплексов молекул РНК и применение для анализа паттернов экспрессии генов в мозге плодовой мушки. Работа проводится на примере изображений плодовой мушки.

Для достижения поставленной цели требуется решить следующие задачи:

1. Изучить методы разделения каналов в экспериментальных биологических изображениях и подобрать пригодные для тестирования в имеющихся данных. Проверить работу методов на тестовых данных из соответствующих статей.
2. Модифицировать и запрограммировать отобранные методы для процедуры обработки имеющихся данных по экспрессии генов в мозге плодовой мушки, выделить настроечные параметры.
3. Получить количественные данные по экспрессии генов в мозге плодовой мушки по имеющимся изображениям.
4. Проанализировать различия в экспрессии генов для разных условий.

# ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ И ОПИСАНИЕ ИСХОДНЫХ ДАННЫХ

## 1.1. Описание исходных данных

Исходные данные представляют собой трехмерные многослойные двух-канальные изображения полученные с помощью конфокального микроскопа. В наборе 5 изображений модельной породе мушки дрозофиллы R338 (под номерами M1, M4, M5, M6, M7), а также 4 изображения дикой породе Sz139.

Современные конфокальные микроскопы обычно имеют несколько фотоприемных каналов, благодаря которым можно получить изображения одновременно в нескольких спектральных областях, т.е. использовать несколько флуорохромов. [4]

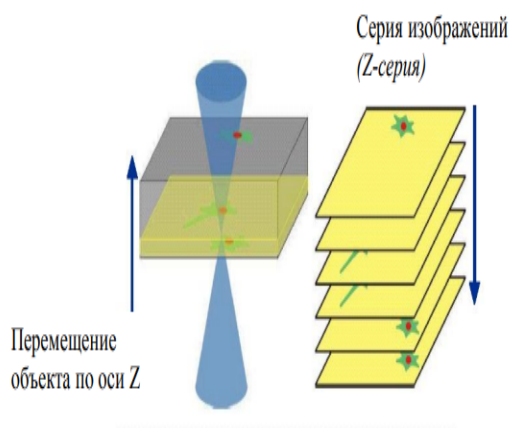


Рис.1.1. Получение серии оптических срезов(Z-серия).

c:1/2 z:60/100 (Ch2-T1); 425.10x425.10  $\mu\text{m}$  (1024x1024); 16-bit; 400MB

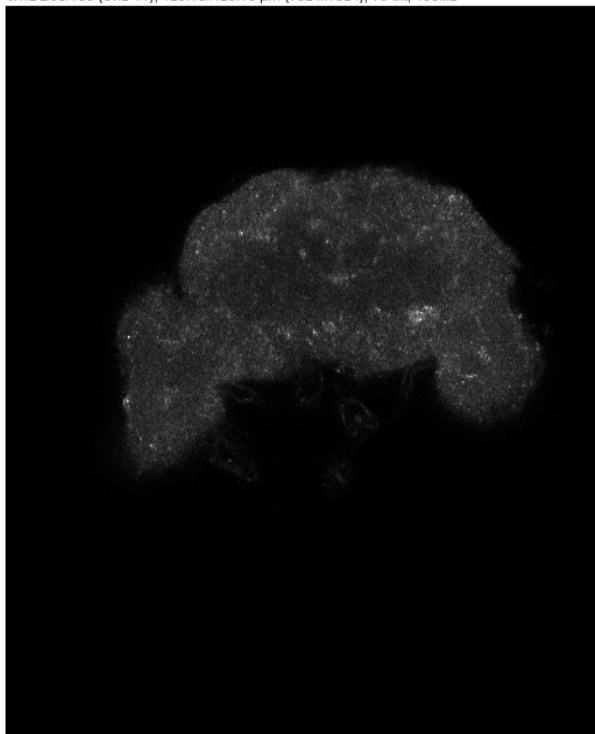


Рис.1.2. Пример трехмерного двухканального изображения мозга плодовой мушки.

## 1.2. Сложность задачи

Существует некоторая проблема в решении поставленной задачи. В исходных изображениях наблюдается паразитное свечение из одного канала микроскопа в другом, что вредит выделению частиц. Данное явление называется автофлуоресценцией. Существует несколько методов решения этой проблемы, которые позволяют уменьшить этот эффект, однако конкретный метод и параметры надо тестировать с конкретными изображениями. Также может понадобиться проводить предобработку и модифицировать последующие шаги всей процедуры.

### 1.3. Существующие методы решений

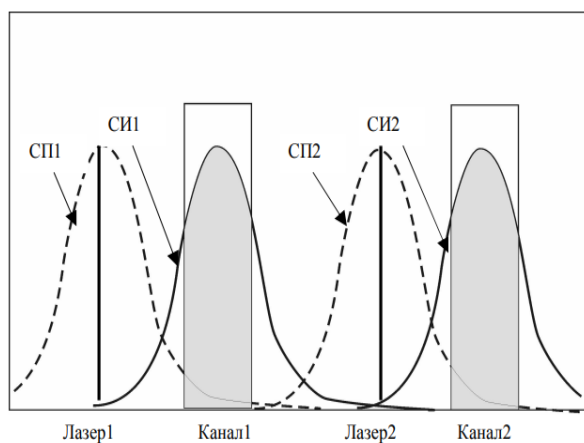


Рис.1.3. Перекрывание спектров полностью отсутствует. СП – спектры поглощения, СИ – спектры испускания флуорохромов 1 и 2.

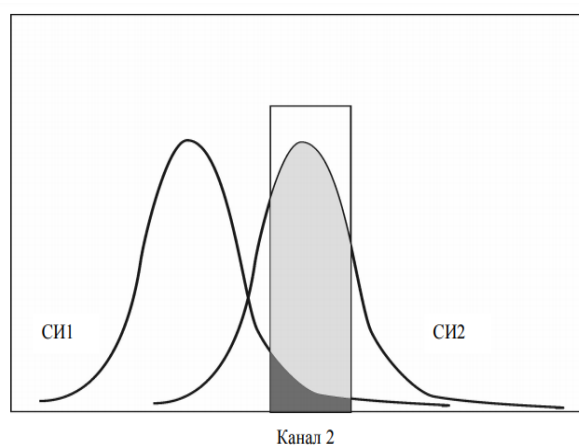


Рис.1.4. Слабое перекрытие спектров. Обозначения те же.

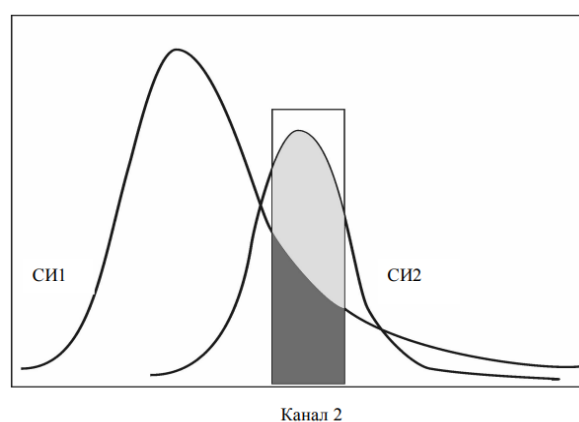


Рис. 4.7. Сильное перекрытие спектров. Обозначения те же.

Рис.1.5. Сильное перекрытие спектров. Обозначения те же

Существуют следующие возможные случаи взаимодействия сигналов от двух флуорохромов. Наилучший вариант показан на Рис.1.3 Перекрытия спектров нет.

На Рис. 1.4 представлен вариант со слабым перекрытием спектров. Часть спектра испускания первого флуорохрома попадает во второй фотоприемный канал. Для этого случая нужна небольшая предобработка. Уменьшить перекрытие можно путём уменьшения мощности первого лазера. Чтобы яркость изображения не падала, нужно усилить первый канал, а также сдвинуть полосу приема второго канала вправо.

В последнем случае - когда перекрытие сильное (Рис. 1.5) необходимо применять последовательное сканирование. То есть сначала включить лазер и фотоприемник для первого канала, затем отключить и повторить для второго канала (режим Multitrack на LSM).

Также существует программный способ уменьшения перекрытия спектров. Он основан на различных математических алгоритмах, учитывающих информацию о спектрах применяемых красителей, применяющих методы линейной алгебры, адаптивном или ручном разделении изображений по приемным каналам.[4] Именно такой способ будет рассмотрен в данной работе.

Наиболее эффективным способом избежания перекрытия спектров является последовательное сканирование, однако у этого подхода есть несколько ограничений. Он требует использования специализированного оборудования и запатентованного программного обеспечения, что является ограничивающим фактором в его широком использовании. Также, например, получения изображений для каждого фотоприемного канала значительно увеличивает время получения изображений и объем данных. [5]

## ГЛАВА 2. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ МЕТОДОВ

### 2.1. Методы обработки изображений реализованные в пакете ProStack

В данной работе производится модификация и улучшение методов обработки изображений реализованных в пакете ProStack.[7]

В пакете ProStack реализованы стандартные и проблемно-ориентированные методы обработки изображений а также методы для получения количественных данных из изображений, полученных на световом или конфокальном микроскопе. Пакет имеет графический интерфейс, для построения сложных сценариев. [8]

Механизм обработки изображения в данном пакете представляет из себя своего рода конвейер - множество зависимых друг от друга различных операций, записанных в один сценарий.

Все методы в рамках пакета разделены на десять классов.[8]

- Комбинирование (Получение одного изображения из нескольких входов)
- Выделение объектов
- Корректировка (Повышение качества изображения)
- Сегментация (Разделение изображения на части/зоны)
- Восстановление
- Морфология (Морфологические операции)
- Геометрия (Изменение свойств изображений)
- Преобразование
- Арифметика (Алгебраические операции)
- Разное

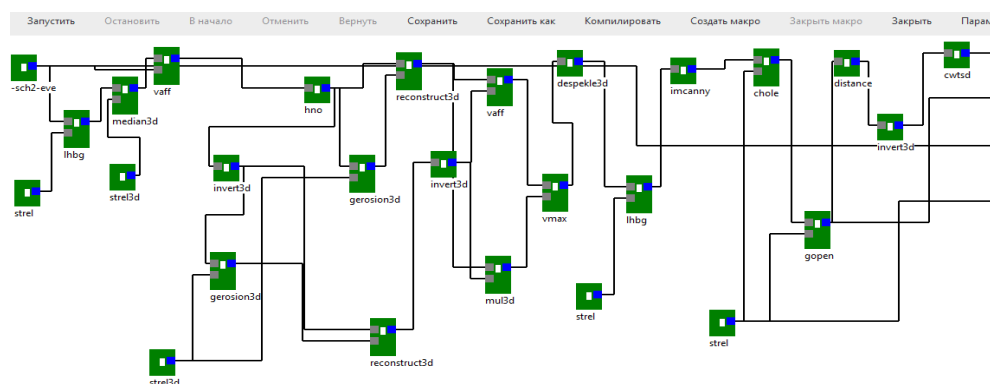


Рис.2.1. Графический сценарий обработки трехмерного изображения из пакета ProStack



Рассмотрим некоторые морфологические операции, которые реализованы в пакете, а также применялись для извлечения количественных данных из изображений мозга мушки.

Для улучшения сегментации (например обработки фона) применяют операцию морфологического размыкания - комбинацию операций эрозии и наращивания. Рассмотрим операции на простом примере. Допустим мы имеем следующее изображение и структурный элемент: [2]

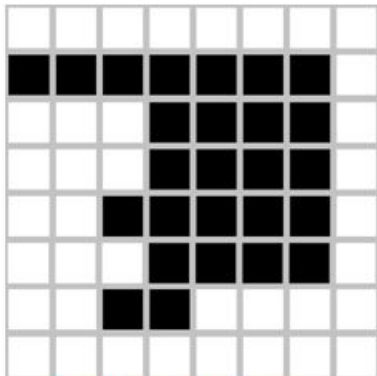


Рис.2.2. Изображение I

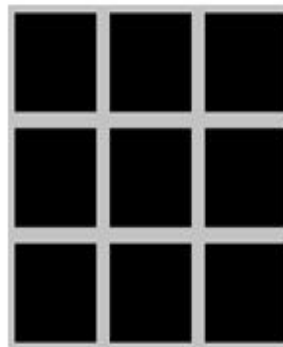


Рис.2.3. Структурный элемент S

**Наращивание** — Структурный элемент "пробегаёт" по всем пикселям бинарного изображения. Если начало координат структурного элемента совпадает с пикселем изображения, то производится логическое сложение структурного элемента с пикселями изображения. Результат записывается в выходное изображение. [2]

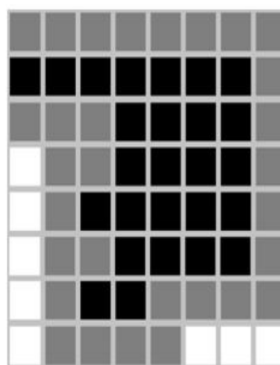


Рис.2.4. Наращивание изображения I структурным элементом S

**Эрозия** — Структурный элемент также "пробегаёт" по всем пикселям изображения. При этом, если каждый пиксель структурного элемента совпадает с пикселями изображения - происходит логическое сложение пикселя находящегося по центру структурного элемента с пикселем изображения. Результат логического сложения также записывается в выходное изображение. [2]

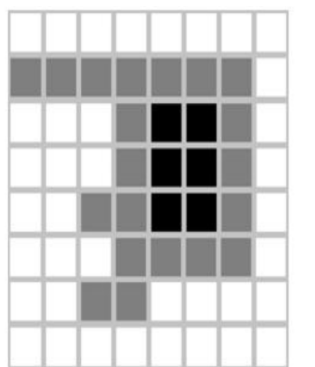


Рис.2.5. Эрозия бинарного изображения  $I$  структурным элементом  $S$

**Размыкание** — эрозия хороша тем что позволяет избавляться от малых объектов представляющих из себя шум. Также из-за этой операции размеры объектов уменьшаются. Это часто неприемлемо для задачи, поэтому после эрозии применяют операцию наращивания используя тот же структурный элемент.[3]

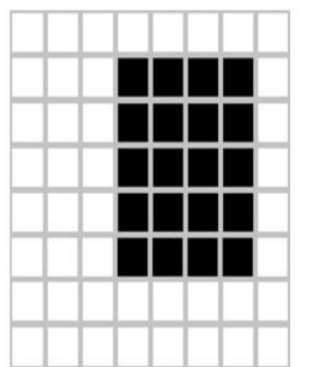


Рис.2.6. Размыкание бинарного изображения  $I$  структурным элементом  $S$

Можно сделать в другом порядке - получится операция **Замыкания**. В таком случае наращивая изображение можно "заполнить" щели. Но чтобы избежать увеличение изображения - производится последующее применение операции эрозии. [3]

Далее рассмотрим довольно распространенную операцию в обработке изображений - **выделение границ**. Данная операция, а именно оператор Кэнни был применен в этой работе для выделения комплексов молекул РНК.

Границей называется изменение яркости на изображении. Она проходит между двумя отличающимися по интенсивности областями. Выделение границ позволяет получить количественные данные из изображений о количестве, площади, размерах областей/зон. Для обнаружения границ могут быть использованы маски.[9]

Рассмотрим операторы *Робертса*, *Собеля*, *Превитта* и алгоритм *Кэнни*.

## Фильтрация

Дадим определение **разрывности** - резкое изменение значений интенсивности. Довольно общим методом поиска разрывности является использование скользящей маски, представляющей из себя обычно квадратную матрицу (или матрица коэффициентов). Применение маски к изображению называют фильтрацией. [1] Схема применения маски показана на рисунке 2.7:

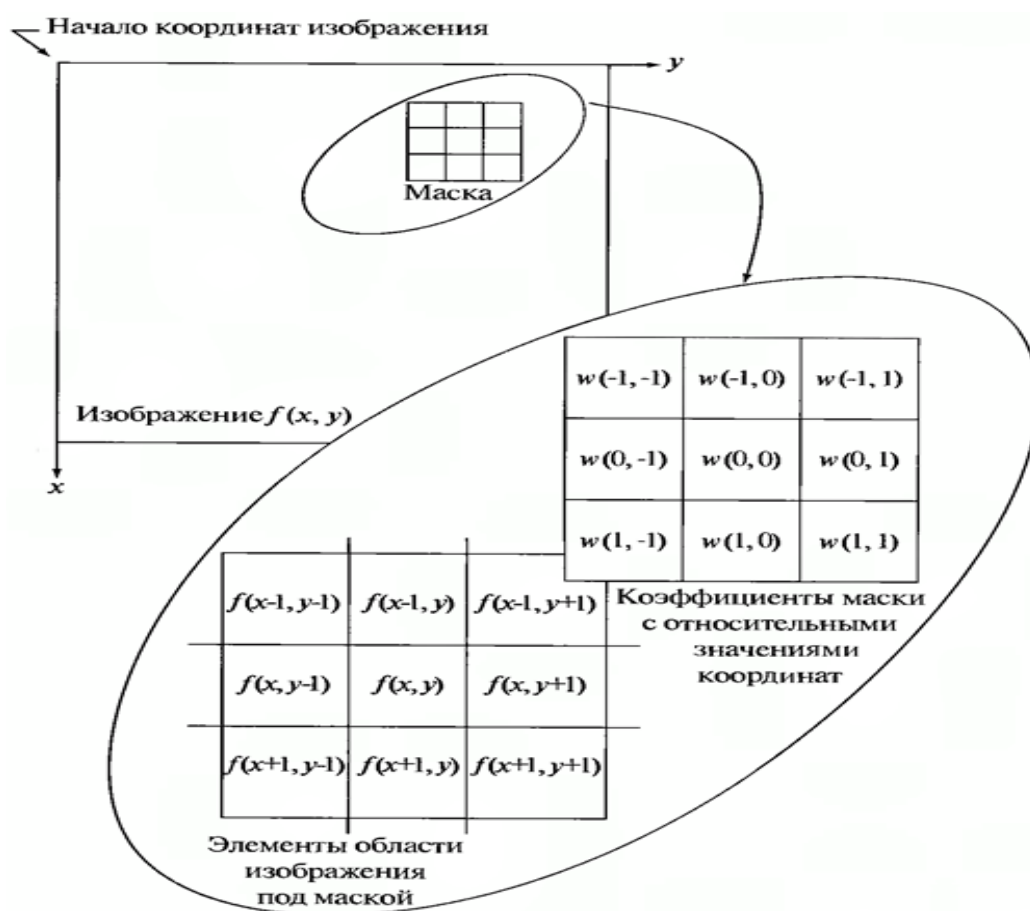


Рис.2.7. Фильтрация изображения маской коэффициентов

Применение маски основано на скольжении маски вдоль изображения по вертикали и горизонтали и вычислении некоторой величины  $R$  - сумма произведения значений каждого пикселя в области, покрытой в некоторый момент маской на соответствующие значения коэффициентов маски. [1] Для примера на рисунке 2.7, значение  $R$  в точке  $(x, y)$  вычисляется как:  $R = w(-1, -1) * f(x - 1, y - 1) + w(-1, 0) * f(x - 1, y) + \dots + w(0, 0) * f(x, y) + \dots + w(1, 0) * f(x + 1, y) + w(1, 1) * f(x + 1, y + 1)$

Для определения разрывов используют аналоги производных и градиента. Производная первого порядка функции  $f(x)$  определяется так:  $\frac{df}{dx} = f(x + 1) - f(x)$ . Вторая :  $\frac{df^2}{dx^2} = f(x + 1) - f(x - 1) - 2f(x)$ . Градиент  $f(x, y)$  в точке  $(x, y)$ :

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}.$$

Для поиска границ объектов вычисляется модуль градиента  $|\nabla f|$ , который равен

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}.$$

### Оператор Робертса

Предположим, что матрица размером 3x3, показанная на рисунке 2.8 представляет пример участка изображения, элементы являются значениями интенсивности.[1]

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

Рис.2.8. Окрестность 3x3

Определение частных производных первого порядка для оператора Робертса:  $G_x = z_9 - z_5$  и  $G_y = z_8 - z_6$  Таким образом можно обработать всё изображение с помощью оператора Робертса описываемого матрицами на рисунке 2.9 и воспользоваться процедурой фильтрации.[1]

-1	0	0	-1
0	1	1	0

Рис.2.9. Маски оператора Робертса

Для масок размеров 2x2 из-за отсутствия центрального элемента ухудшается результат выполнения фильтрации. Данный минус компенсируется высокой скоростью обработки всего изображения.

### Оператор Превитта

Оператор Превитта тоже работает с областью 3x3, но градиенты вычисляются иначе:  $G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$  и  $G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$  Данные формулы описываются масками на рисунке 2.10.

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Рис.2.10. Маски оператора Превитта

### Оператор Собеля

Оператор Собеля в отличие от оператора Превитта использует весовой коэффициент равный двум для средних элементов:  $G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$  и  $G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$

Данное увеличение коэффициентов направлено на уменьшение эффекта сглаживания.[1]

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Рис.2.11. Маски оператора Собеля

Далее по полученным значениям  $G_x$  и  $G_y$  после применения масок вычисляется вектор градиента.  $|\nabla f| = \sqrt{G_x^2 + G_y^2}$ . Решение о перепаде интенсивности, то есть наличии границы применяется после сравнение модуля градиента с некоторым пороговым значением (подбирается эмперически).[1]

### Детектор границ Канни

Детектор Кэнни является одним из самых популярных алгоритмов поиска границ. Ключевым этапом является подавление шума на контурах, что значительно может повлиять на результат выделения.

Шаги алгоритма детектора границ Кэнни:

- Размытие изображения.
- Поиск градиента.
- Подавление немаксимумов.
- Трассировка границ.

Алгоритм Канни по сравнению с рассмотренными выше операторами менее восприимчив к шуму на изображении.[9]

## 2.2. Инструмент поиска автофлуоресценции. AFid

Данный метод является программным способом удаления автофлуоресценции 1.2. То есть применяется после получения изображения и делает его весьма практичным так как пользователям не нужно менять свои экспериментальные процедуры.[5]

### 2.2.1. Входные данные и требования

Алгоритм принимает на вход два изображения - двухканальное изображение исследуемого объекта, разделенное на два канала. Для разбиения использовался пакет Fiji. [10]

Важным требованием алгоритма является то что автофлуоресценция не должна перекрываться реальным сигналом. То есть не должно быть фонового перекрытия между реальным сигналом для конкретного канала и сигналом являющегося автофлуоресцентным.[5]

### 2.2.2. Создание маски пересечения

Маска пересечения двух каналов используется для исключения автофлуоресценции. Она содержит только те сигналы, которые присутствуют в обоих каналах, и поэтому содержит интересные автофлуоресцентные области. Маска строится следующим образом: к каждому из каналов изображение применяется Гауссово размытие с  $\sigma = 2$  и с размером Гауссова ядра  $k$ , рассчитываемого по формуле  $k = 2 * \text{ceil}(2 * \text{sigma}) + 1$ , где  $\text{ceil}$  - функция округления вверх. Далее к размытым каналам применяется порог Оцу, которое определяет оптимальное глобальное пороговое значение из гистограммы изображения, и получаются две бинарные маски для каждого канала. Результирующая маска пересечения получается путем применения друг к другу "логического И" бинарных масок полученных на предыдущем шаге. [5]

### Алгоритм генерации маски пересечения

**Input:** Первый канал - ch1, второй канал - ch2

**Output:** Маска пересечения - result mask

1.  $\sigma = 2$ ;
2. result of blurred channel 1 = `imgaussfilt(ch1, sigma)`;
3. result of blurred channel 2 = `imgaussfilt(ch2, sigma)`;;
4.  $th1 = \text{threshold}(\text{result of blurred channel 1}, cv.THRESH\_OTSU)$ ;
5.  $th2 = \text{threshold}(\text{result of blurred channel 2}, cv.THRESH\_OTSU)$ ;
6. result mask =  $th1 \ \& \ th2$ ;
7. **return** result mask;

### 2.2.3. Кластеризация для идентификации автофлуоресценции

В соответствии с маской пересечения объекты на каждом из каналов изображения делятся на регионы. Для применения к изображениям мозга мушки это могут быть комплексы молекул РНК. Далее вычисляются характеристики для найденных зон каждого канала. Характеристики включают стандартное отклонение, эксцесс (четвертый центральный момент, деленный на квадрат дисперсии), а также межканальный коэффициент корреляции Пирсона значений интенсивности соответствующих пикселей. Далее эти характеристики были преобразованы: путем натурального логарифма для стандартное отклонения и эксцесса, с помощью обратного гиперболического тангенса для линейной корреляции. Все преобразования были стандартизированы путем деления на стандартное отклонение преобразованных значений признаков.

Далее выполняется кластеризация преобразованных и стандартизованных значений характеристик для определения кластера тех регионов, которые, вероятно, будут автофлуоресцентными. Кластер с самым высоким средним значением корреляции был определен как кластер, содержащий автофлуоресцентные области. Важно подобрать правильное число кластеров для обнаружение автофлуоресцентных областей.

Также возможен вариант с автоматизированным выбором оптимального числа кластеров  $k$ . Итак, вариант алгоритма описанный выше, без определения оптимального числа кластеров повторяется для  $k$  от 3 до 20. При каждой итерации определяется уже два класса с наибольшими средними значениями корреляции и рассчитываются значения t-критерия Стьюдента для двух соответствующих классов

выборки значений корреляций. Далее значения  $t$ -критерия наносятся на график относительно  $k$ , график представляет собой асимптотически убывающую функцию (см. рис. 2.12)

Определение оптимального числа кластеров по графику производится следующим образом: проводится прямая линия, соединяющая статистическое значение для самого низкого и высокого  $k$ . Измеряется расстояние перпендикуляра проведенного от каждой нанесенной точки к линии, тогда оптимальное  $k$  соответствует точке с наибольшим расстоянием. Данный способ проиллюстрирован на рисунке 2.12. Точка помеченная красной звездочкой соответствует оптимальному числу кластеров.[5]

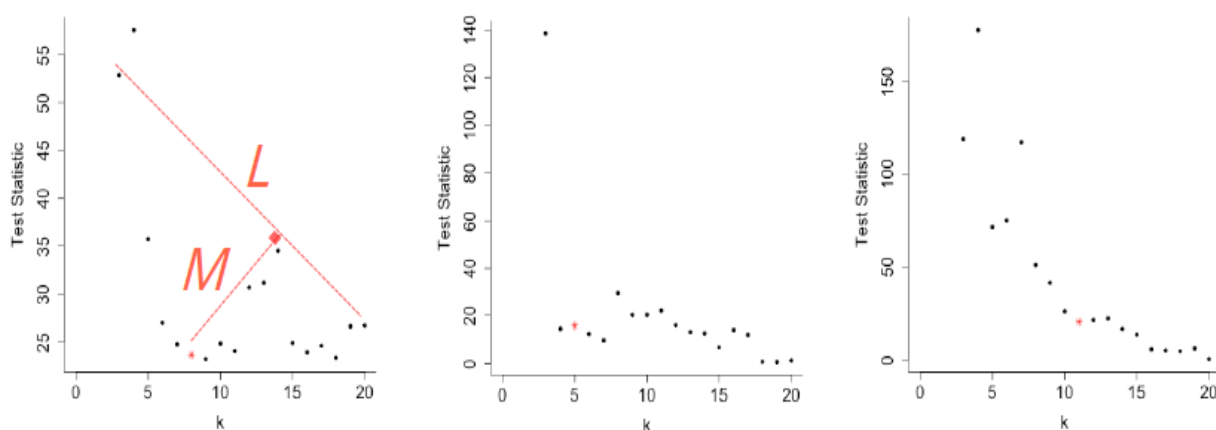


Рис.2.12. Определения оптимального числа кластеров через  $t$ -критерий.

Далее результирующая маска с автофлуоресцентными объектами получается из маски пересечения, в которой сохранились лишь те регионы идентифицированные как автофлуоресценция.

#### 2.2.4. Расширение автофлуоресцентных областей

После кластеризации и получения маски автофлуоресцентных объектов применяется функция расширений областей автофлуоресценции. Наличие этой процедуры объясняется тем что найденные до кластеризации регионы не всегда охватывают нужные области из-за неточных порогов, поэтому нужны удалить дальнейшее "свечение". Суть данного метода состоит в том, чтобы равномерно распределить точки внутри автофлуоресцентного объекта, а затем расширяться от этих точек во всех направлениях, пока не будет выполнено условие остановки.



Дадим определение скелету изображения: *Скелет* - множество точек-центров всех вписанных кругов фигуры.

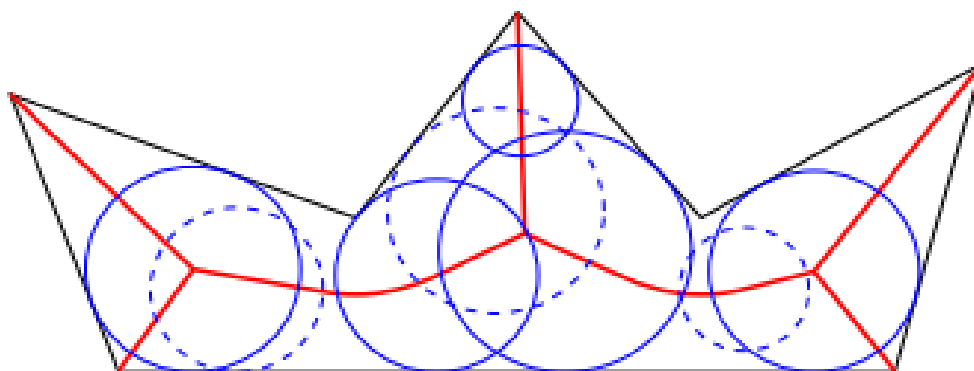


Рис.2.13. Скелет - красная кривая, вписанные круги фигуры - синие.

Сначала строится скелет маски автофлуоресцентных объектов, далее равномерно распределяются точки по построенному скелету (каждые 20 пикселей). Затем происходит расширение от этих точек до тех пор, пока градиент яркости пикселей от границы области не начнет увеличиваться, указывая на конец или начало соседнего объекта.[5]

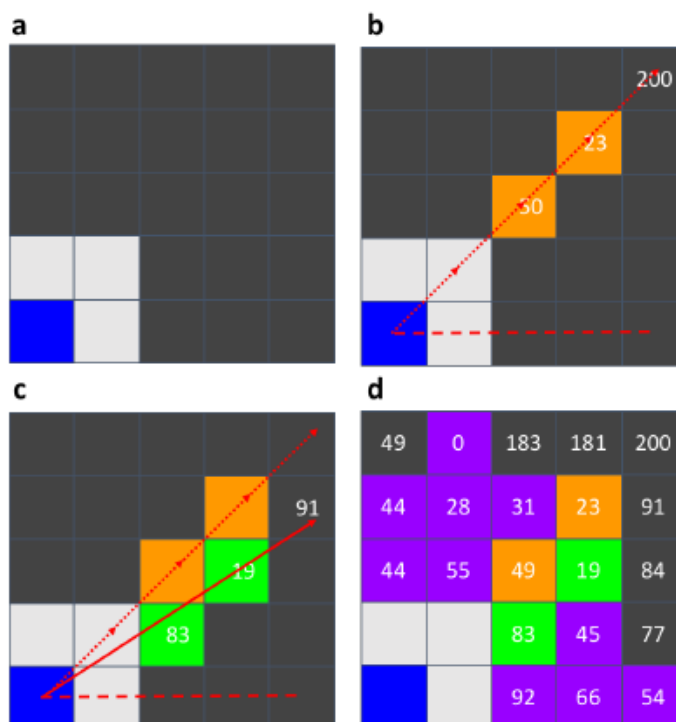


Рис.2.14. Схема иллюстрирующая шаги алгоритма расширения объектов.

На рисунке 2.14 показано, как происходит расширение автофлуоресцентных объектов. Каждый квадрат это пиксель изображения. Белые пиксели - пиксели принадлежащие автофлуоресцентному объекту. Синий пиксель - точка откуда может происходить расширение. Темно-серые пиксели обозначают пиксели, не являющиеся автофлуоресцентным объектом. От точки расширения (синий пиксель) проводится прямая линия. Как только линия достигает края автофлуоресцентного объекта (белые пиксели), она начинает измерять значение пикселей, которые она пересекает. Линия будет продолжать расширяться наружу до тех пор, пока значение следующего пикселя меньше или равно предыдущему значению пикселя. Все пиксели, удовлетворяющие этому условию, закрашиваются оранжевым цветом. Новые расширенные пиксели, закрашенные прямой линией исходящей под другим углом, обозначены зеленым и фиолетовыми цветами. Все цветные пиксели теперь образуют новый расширенный объект, который изначально частично состоял из 4 пикселей (левый верхний рисунок а).[5]

### 2.2.5. Обзор алгоритма

Шаги алгоритма приведены на рисунке 2.15.

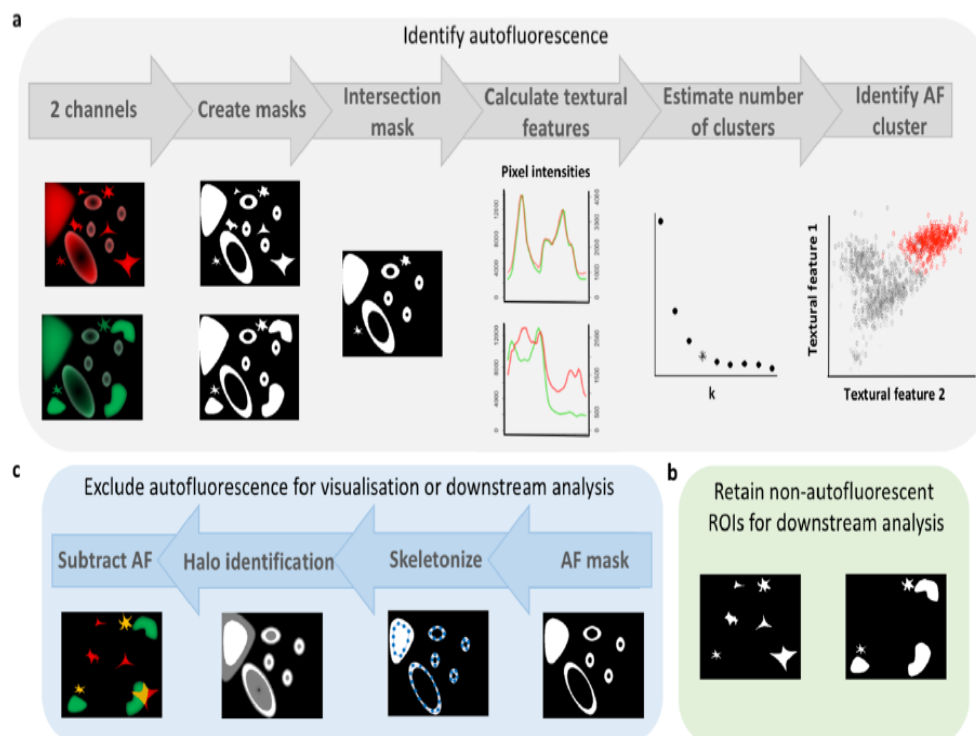


Рис.2.15. Иллюстрация шагов алгоритма AFid.

(ДОПИСАТЬ ПСЕВДОКОД И ОПИСАНИЕ ШАГОВ) .....

## ГЛАВА 3. ПРИМЕНЕНИЕ ОПИСАННЫХ МЕТОДОВ

### 3.1. Модификация AFid

#### 3.1.1. Переписывание на Python

Алгоритм идентификация автофлуоресцентных объектов AFid был выложен разработчиками в открытом доступе и реализован в пакете Matlab. Из-за некоторых трудностей внедрения приложения написанного на языке требующей лицензии в пакет ProStack, и для доступности использования модификаций для всех пользователей - решено было переписать AFid на языке Python (3.6).

Методы из библиотеки "Image Processing Toolbox" пакета Matlab, которые использовались в AFid были заменены методами из библиотек skimage, cv2, scipy и sklearn языка Python.

#### 3.1.2. Предобработка входных данных

Одним из требований инструмента AFid было отделение автофлуоресценции от реального сигнала 2.2.1. Но не для всех исходных данных в данной работе данное требование выполнялось. В некоторых изображениях мозга мушки, в одном из каналов наблюдалась почти однородная картина, где комплексы молекул тяжело отделялись, а в другом канале могли быть хорошо отделены.(см. рис 3.1)

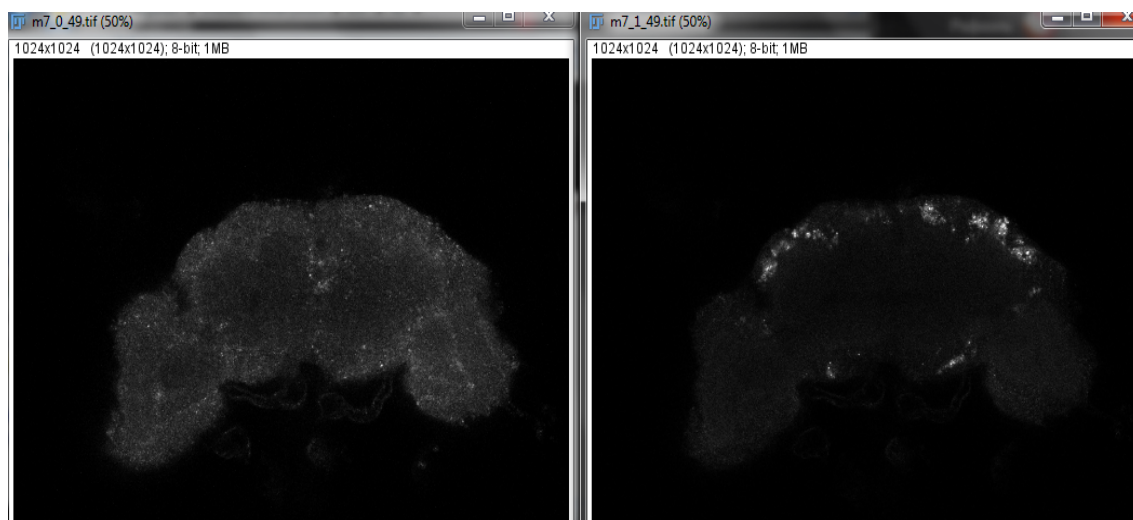


Рис.3.1. Серединный срез многослойного трехмерного двухканального изображения R338 M7 мушки дрозофиллы.

В этом случае алгоритм AFid очень неточно генерировал маску пересечения, она получалась слишком однородной, без отделенных сигналов благодаря большому количеству яркого фона в одном из каналов.(см. рис. 3.2)

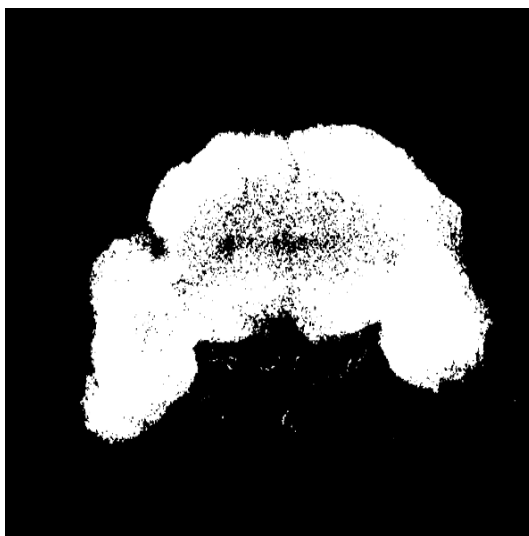


Рис.3.2. Маска пересечения до удаление фона.

Было принято решение реализовать предобработку входных изображений - удалить фон мешающий идентификации областей автофлуоресценции и реальных сигналов. Удаление фона имеет следующую процедуру:

Расчитывается некоторое пороговое значение  $d$  определяемое как

$$d = meanvalue + coeff * std$$

Тут  $meanvalue$  - среднее значение пикселей изображения,  $coeff$  - эмперически подобранный коэффициент равный 0.8,  $std$  - стандартное отклонений значений пикселей изображения.



Рис.3.3. Маска пересечения после удаления фона.

После удаления фона результат выглядит получше, теперь результат умножения масок больше похож на правду, т.к на маске присутствуют объекты которые имеют большую яркость в обоих каналах а не только в первом.(см. рис. 3.3)

### **3.2. Применение AFid**

Алгоритм AFid применялся к исходным данным описанным в 1.1 (5 изображений модельной мушки и 4 дикой).

После обработки инструментом обнаружения и удаления автофлуоресценции, изображения обрабатывались сценарием пакета Prostack для выделения комплексов молекул РНК. Затем результаты сравнивались с теми, которые были получены сценарием без применения AFid.

### **3.3. Проверка статистической гипотезы**

После применения инструмент AFid идентифицирует и удаляет автофлуоресцентные объекты. Следовательно паразитное свечение из одного канала в другом должно исчезнуть. Было решено проверить двухсторонний критерий Вилконсона о наличии следующей гипотезы: медиана смещений среднеквадратичных отклонений значений пикселей выделенных комплексов молекул РНК после применения AFid должна быть отлична от нуля.

Для проверки критерия, были подготовлены необходимые данные - количественные результаты экспрессии, полученные сценарием ProStack и записанные в файл .csv. Критерий проверялся для пяти изображений модельной мушки и четырех для дикой породы, по очереди для двух каналов.

### **3.4. Кластеризация мозга мушки**

Полученные количественные данные сценарием ProStack были кластеризованы методом KMeans, предварительно построены гистограммы значений интенсивности пикселей комплексов молекул РНК: первая гистограмма - учитывала частоту встречаемости пикселей по мере увеличения значения интенсивности, вторая - количество комплексов молекул для каждого слоя изображения. Гистограммы необходимы для удаления вероятного шума, который в изображениях мозга мушки присутствует в первых либо последних слоях. Также по гистограмме можно

выловить шум убирая те выделенные объекты у которых значения интенсивности пикселей очень малое или сильно меньше остальных.

### **3.5. Название параграфа**

## ГЛАВА 4. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ

### 4.1. Результат проверки статистической гипотезы

Критерий Вилконсона показал результат  $p\text{-value}$  больше 0.1 для модельной мушки дрозофиллы и чуть больше 0.5 для дикой породы. Значит можно сказать что никакой статистической значимости в изменении стандартных отклонений нет.

### 4.2. Результаты кластеризации и фильтрации

#### 4.2.1. Результат фильтрации

Ниже приведены построенные гистограммы для удалени шума изображения модельной мушки М4.

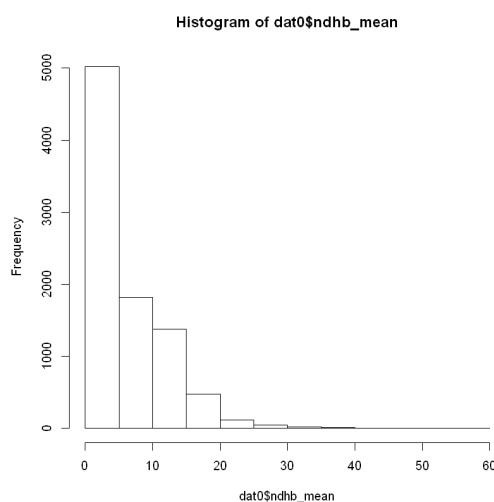


Рис.4.1. Гистограмма частоты встречаемости пикселей по мере увеличения значения интенсивности для изображения М4.

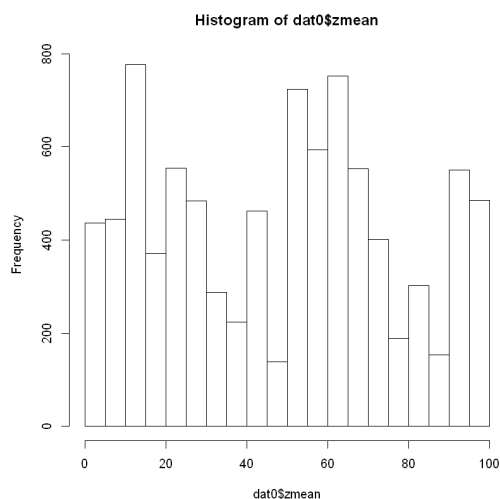


Рис.4.2. Гистограмма показывающая количество комплексов молекул для каждого слоя изображения М4.

После удаления объектов подозрительных на шум (крайние объекты во второй гистограмме и имеющие очень малую интенсивность в первой) были получены следующие результаты фильтрации.

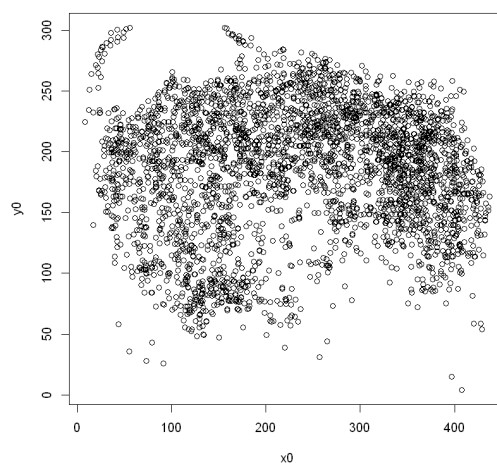


Рис.4.3. Результат фильтрации для изображения мозга модельной мушки М4.



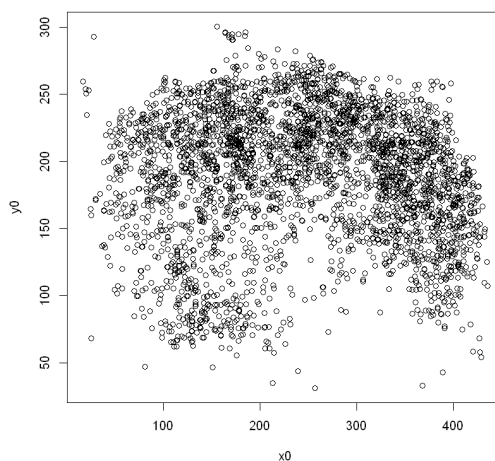


Рис.4.4. Результат фильтрации для изображения мозга модельной мушки M4 после применения инструмента удаления автофлуоресценции AFid.

По результатам фильтрации можно сказать что после применения AFid изображение мозга лучше кластеризуется. Так, например, на рисунке 4.4 убраны висячие артефакты которые присутствуют в изображении до применения AFid на рисунке 4.3.

## ГЛАВА 5. ЗАКЛЮЧЕНИЕ

В данной работе стояла следующая задача: разработка алгоритма для выделения на экспериментальных изображениях комплексов молекул РНК и применение для анализа паттернов экспрессии генов в мозге плодовой мушки.

По итогам был представлен модифицированный сценарий для выделения комплексов молекул РНК на изображениях, были описаны новые алгоритмы внесенные в сценарий, их результаты работы и модификация.

Были получены количественные результаты экспрессии генов в мозге плодовой мушки.

## ГЛАВА 6. ВЫВОДЫ

На основании проделанной работы можно сделать следующие выводы:

1. Адаптирован инструмент идентификации автофлуоресценции AFid: метод был переписан на язык Python, были внесены модификации в работу алгоритма, изменены параметры применительно для изображений мозга мушки и представлены результаты применения к исходным изображениям.
2. Получены количественные данные экспрессии генов с детекцией автофлуоресцентных объектов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Гонсалес Р. В. Р.* Цифровая обработка изображений М: Техносфера: дис. . . . канд. / Гонсалес Р. Вудс Р. — 2005. — 1007 с.
2. *Горьков А.* Математическая морфология //. — URL: <https://habr.com/ru/post/113626/> (дата обращения: 12.02.2011).
3. *Шапиро Л. Д.* Компьютерное зрение. изд / М.: БИНОМ. Лаборатория знаний. — 2006. — 752 с.
4. *Штейн Г. И.* Руководство по конфокальной микроскопии - СПб: ИНЦ РАН, //. — Стандартинформ, 2007. — (Сер.: ил.ISBN).
5. *Baharlou H. C. N. P. Bertram K. M., Sandgren K. J., Cunningham A. L., Harman A. N., Patrick E* AFid: A tool for automated identification and exclusion of autofluorescent ob-jects from microscopy images. bioRxiv, 566315 //. — 2019. — URL: <https://doi.org/10.1101/566315>.
6. *Bradski G.* The OpenCV Library. — 2000. — P. 122–125.
7. *K. K. et al.,* Quantitative analysis of the heterogeneous population of endocytic vesicles. Journal of Bioinformatics and Computational Biology, 10:1750008: Master's thesis / K. Kozlov. — The school where the thesis was written, 2017.
8. *Kozlov K.N. Baumann P. Waldmann J. et al.* TeraPro, a system for pro-cessing large biomedical image. // Pattern Recognit. Image Anal. — 2013. — P. 23, 488–497. — URL: <https://doi.org/10.1134/S105466181304007X>.
9. *Muthukrishnan R R. M.* International Journal of Computer Science, Information Technology (IJCSIT), 3(6).
10. *Schindelin J. A.-C. I.* Fiji: an open-source platform for biological-image analysis, PMID 22743772: tech. rep. — 2012. — 676-682. — URL: <http://dx.doi.org/10.1038>.