Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики

Работа допущена к защите
Должность руководителя М
_____ И.О. Фамилия
«_____» _____ 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
РАБОТА БАКАЛАВРА
РАЗРАБОТКА МЕТОДА ОПРЕДЕЛЕНИЯ ЭКСПРЕССИИ ГЕНОВ ПО ИЗОБРАЖЕНИЯМ МОЗГА ПЛОДОВОЙ МУШКИ

по направлению подготовки 01.03.02 Прикладная математика и информатика
Направленность (профиль) 01.03.02_04 Биоинформатика

Выполнил
студент гр. 3630102/70401 Р.А. Темиргалиев

Руководитель
должность,
степень, звание К.Н. Козлов

Консультант
должность, степень И.О. Фамилия

Консультант
по нормоконтролю И.О. Фамилия

Санкт-Петербург
2021

СОДЕРЖАНИЕ

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 экспресси 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 органимы(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.

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 трёхмерные многослойные двухканальные изображения полученные с помощью конфокального микроскопа. В наборе 5 изображений модельной породы мушки дрозофиллы R338 (под номерами M1, M4, M5, M6, M7), а также 4 изображения дикой породы Sz139. Размер изображений 1024 на 1024 пикселей, объём - 100 мегабайт.

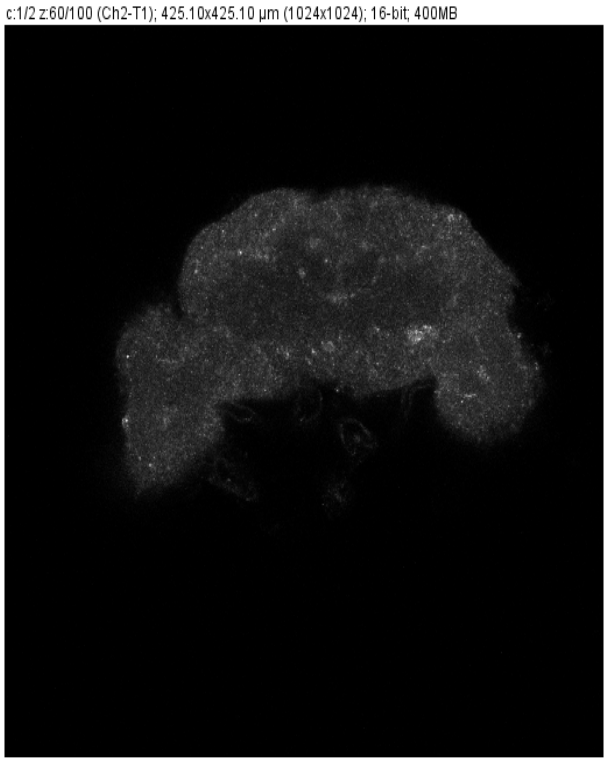

Рис.1.1. Пример трехмерного двухканального изображения мозга плодовой мушки.

1.1.1. Получение данных

Мушки дрозофиллы были выращены при температуре +25 градусов, 12 часов при свете и темноте. Каждый исследуемый пол и генотип отбирались из отдельных популяций. Родителям давали 24 часа на откладку яиц.

Во время препарировании отделялась ткань головы и ротовой аппарат, во избежании нанесения повреждений мозгу.

Для исследования использовали дрозофилл поздней стадии (P15). Дрозофил на поздней стадии куколки легко идентифицировать и препарировать. Также,

центральная нервная система мушки на этой стадии и взрослых самцов имеет малые отличия.

1.1.2. Оборудование для получения исходных данных

Образцы были получены постериорно-антериорном направлении с помощью конфокальной системы Zeiss LSM 780 (Carl Zeiss MicroImaging, Inc., Thornwood, NY), объектив: PlanApochromat 20×/1.40.

Конфокальные микроскопы имеют несколько фотоприемных каналов, что позволяет получить изображения одновременно в нескольких спектральных диапазонах. [4]

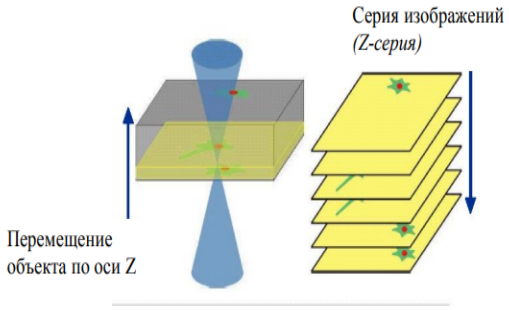

Рис.1.2. Получение серии оптических срезов(Z-серия).

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

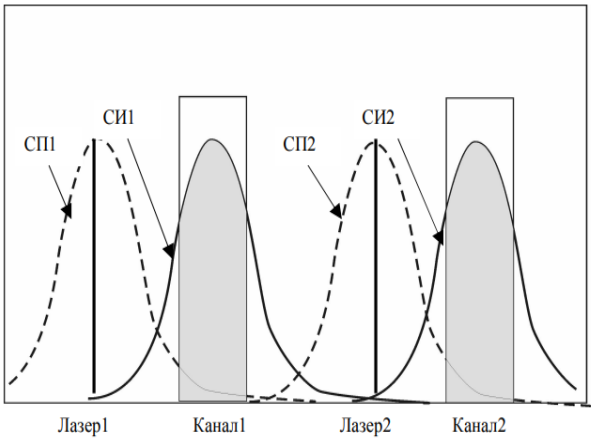

Рис.1.3. Спектры не перекрываются. СП – спектры поглощения, СИ – спектры испускания флуорохромов 1 и 2.

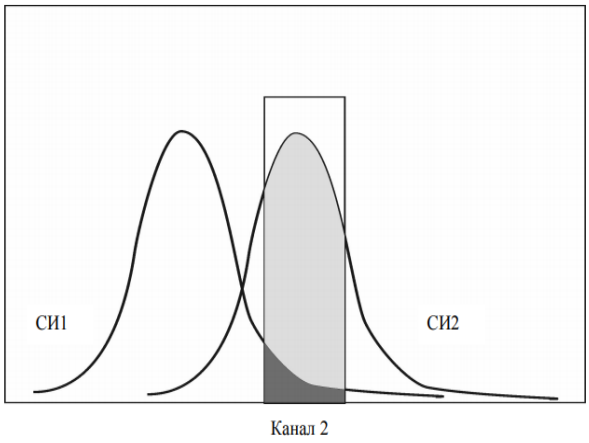

Рис.1.4. Спектры перекрываются слабо.

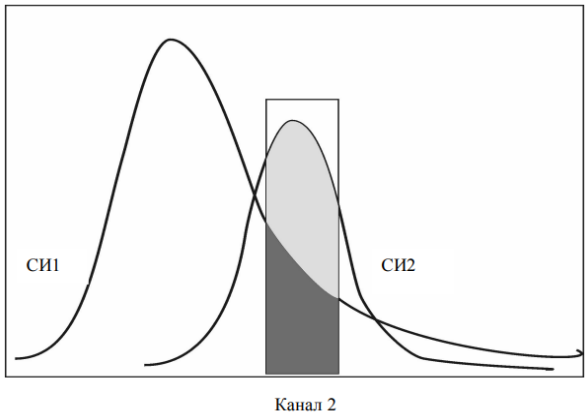

Рис. 4.7. Сильное перекрытие спектров. *Обозначения те же.*

Рис.1.5. Спектры перекрываются сильно.

Рассмотрим все случаи взаимодействия сигналов от флуорохромов. Отсутствие перекрытия показано на Рис.1.3. В этом случае можно сканировать одновременно.[4]

Слабое перекрытие спектров на рис. 1.4. Здесь небольшая часть спектра испускаемого первым флуорохромом попадает в диапазон второго канала. Можно попробовать уменьшить мощность лазера для первого флуорохрома, а также, во избежании потери яркости осуществить сдвиг диапазона приема второго канала правее.[4]

В последнем случаем - когда перекрытие сильное (см. рис. 1.5) необходимо сканировать последовательно. То есть, сначала включить лазер и фотоприемник только для первого канала и для второго их отключить, затем аналогичено включить для второго и выключить для первого. [4]

Также, помимо последовательного сканирования существует вычислительный способ уменьшения перекрытия, основанного различных математических алгоритмах с применением теории по линейной алгебре, математиеской статистике, машинному обучению и статистическому анализу. В данном приёме используется информация о спектрах используемых красителей, значениях интенсивности пикселей изображений.[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 конвеер - 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 свойст изображений)
- Преобразование
- Арифметика (Алгебраические операции)
- Разное

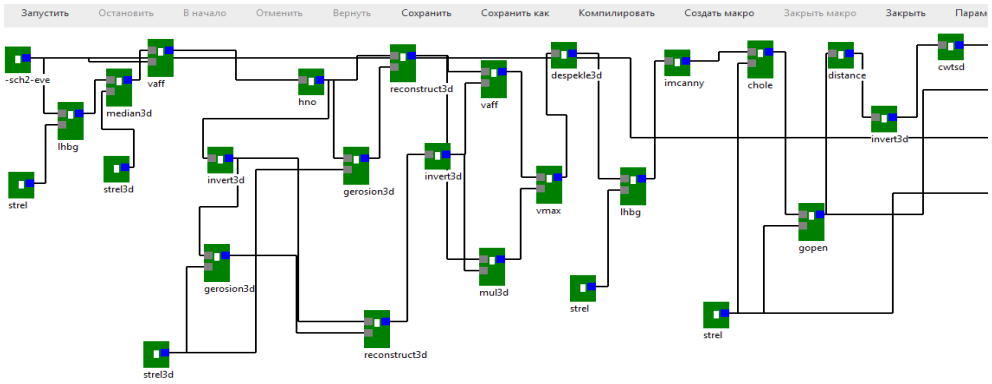

Рис.2.1. Графический сценарий обработки трехмерного изображения из пакета ProStack

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

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

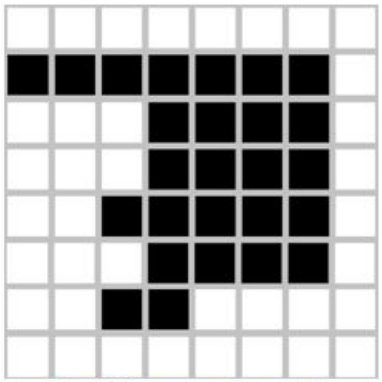

Рис.2.2. Изображение I

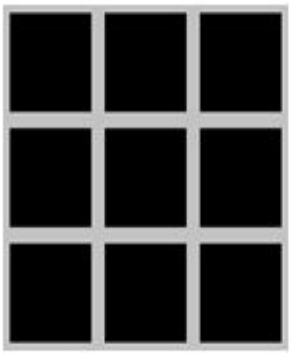

Рис.2.3. Структурный элемент S

Наращивание — Структурный элемент "пробегает"по всем пикселам 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]

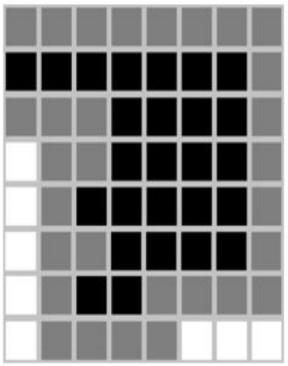

Рис.2.4. Наращивание изображения I структурным элемнтом S

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

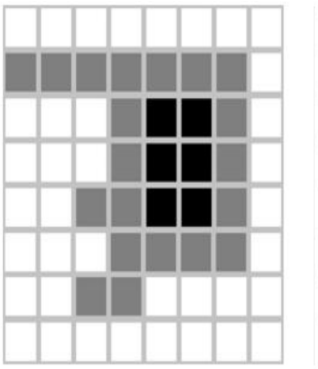

Рис.2.5. Эрозия бинарного изображения I структурным элемнтом S

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

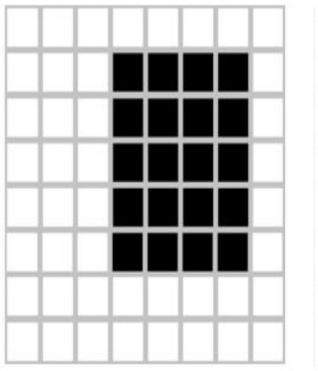

Рис.2.6. Размыкание бинарного изображения I структурным элементом S

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]

Далеее рассмотрим довольно распространенную операцию в обработе 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:

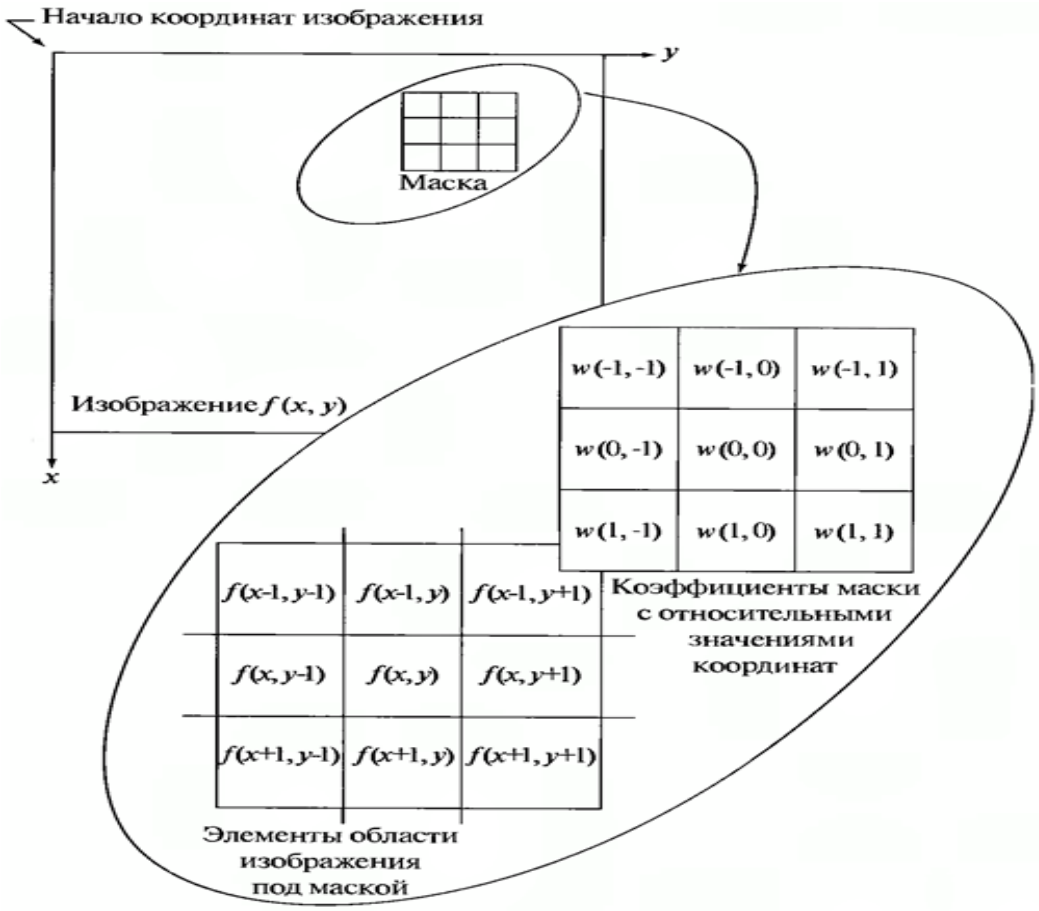

Рис.2.7. Фильтрация изображения маской коэффициентов

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 соотвествующие значения коэффициентов маски.[1] Для примера на рисунке 2.7, значение **R** в точке **(x,y)** вычисляется как: $R = w(-1,-1) * f(x-1, y-1) + w(-1,0) * f(x-1, y) + ... w(0,0) * f(x, y) + ... + w(1,0) * f(x+1, y) + w(1,1) * f(x+1, y+1)$

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):

$$\nabla f = [\frac{G_x}{G_y}] = \frac{\frac{df}{dx}}{\frac{df}{dy}}.$$

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
0	0	0
1	1	1

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

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

Рис.2.11. Маски оператора Собеля

Далее по полученным значениям G_x и G_y после применения масок вычисляетя 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. Он был разработан Джоном Ф. Кэнни в 1986 году. Ключевым этапом является подавление шума на контурах, что значительно может повлиять на результат выделения. [9]

Рассмотрим этапы алгоритма детектора границ Кэнни [6]:

1. *Размытие изображения*. Так как операция выделения границ чувствительна к шуму в изображении, то необходимо этот шум удалить - с помощью гауссовского фильтра.
2. *Поиск градиента яркости изображения*. Далее, сглаженное на первом этапе изображения фильтруют оператором Собеля. То есть вычисялется

первая производная по горизонтали G_x и по вертикали G_y. Далее вычисляется вектор градиента и его модуль $|\nabla f| = \sqrt{G_x^2 + G_y^2}$. Направление градиента перпендикулярная границе и расчитывается как $\text{tg}^{-1}(\frac{G_y}{G_x})$.

3. *Подавление не локальных максимумов*. После получения значения модуля и направления для градиента на 2 этапе выполняется удаление пикселей представляющих собой лишние участки границ. То есть, рассматриваются значения пикселей границы в направлении градиента, и удаляются пиксели, значение интенсивности которых не превышает значения двух соседних в рассматриваемом в заданном направлении. Таким образом данный этап позволяет получить более тонкие края границ изображения.
4. *Пороговые значения для границ*. На данном этапе идет отсечение некоторых границ по пороговому значению. Например, допустим заданы два порога *minTresh* и *maxTresh*, тогда любые границы со значениями интенсивности больше *maxTresh* обязательно будут порогами, а те что ниже *minTresh* - удаляются. Все остальные границы значения интенсивностей которых лежат между этими порогами классифицируются в зависимости от того связаны ли они с границами интенсивности которых превышают *maxTresh* или нет.

   На рис. 2.12 проиллюстрирована работа совершаемая на данном этапе. Граница A находится выше *maxTresh* и поэтому считается истинной, граница C находится ниже *maxTresh* но связана с A, которая в свою очередь считается границей, в таком случае C также не отсеивается. Но граница B находится ниже *maxTresh* и не имеет связи как C, в таком случае B удаляется. [6]

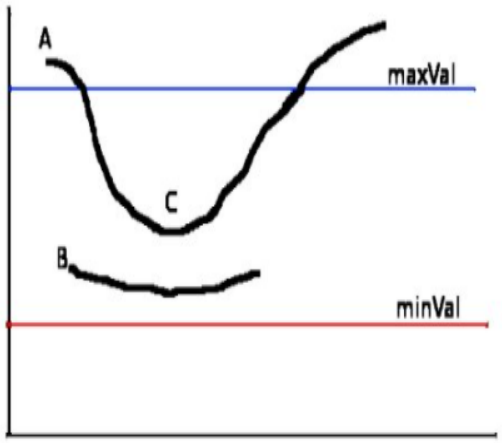

Рис.2.12. Отсечение найденных границ по пороговому значению

2.2. Инструмент поиска автофлуоресценции. AFid

Данный метод является програмным 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 обеих каналах, и поэтому содержит интересующие автофлуоресцентные области. Маска строится следующим образом: к каждому из каналов изображение применяется Гауссово размытие с $\sigma = 2$ и с размером Гауссово ядра k, расчитываемого по формуле

$k = 2 * ceil(2 * sigma) + 1$, где *ceil* - функция округления вверх. Далее к размытым каналам применяется порог Оцу, которое определяет оптимальное глобальное пороговое значение из гистограммы изображения, и полуются две бинарные маски для каждого канала. Результирующая маска пересечения получается путем применения друг к другу "логического И"бинарных масок полученных на предыдущем шаге. [5]

Псевдокод алгоритма генерации маски пересечения

Input: Первый канал - ch1, второй канал - ch2

Output: Маска пересечения - result mask

```
1. sigma = 2;
2. blurred chanel 1=imgaussfilt(ch1) //Гауссово размытие 1 канала;
3. blurred chanel 2=imgaussfilt(ch2) //Гауссово размытие 2 канала;
4. th1 = threshold(blurred chanel 1) //Применение порога Оцу;
5. th2 = threshold(blurred chanel 2);
6. resmask = th1 & th2 //логическое И бинарных масок для каждого канала;
7. return resmask;
```

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. Затем производится пороговая фильтрация площадей регионов. Все регионы, площади которых меньше порогового - отбрасываются. 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, вероят-

но, 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 расчитываются значения t-критерия Стьюдента для двух соответствующих классов выборок значений корреляций. Далее значения t-критерия наносятся на график относительно *k*, график представляет собой асимптотически убывающую функцию (см. рис. 2.13)

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 проиллюстрировн на рисунке 2.13. Точка помеченная красной здвездочкой соотвествует оптимальному числу кластеров.[5]

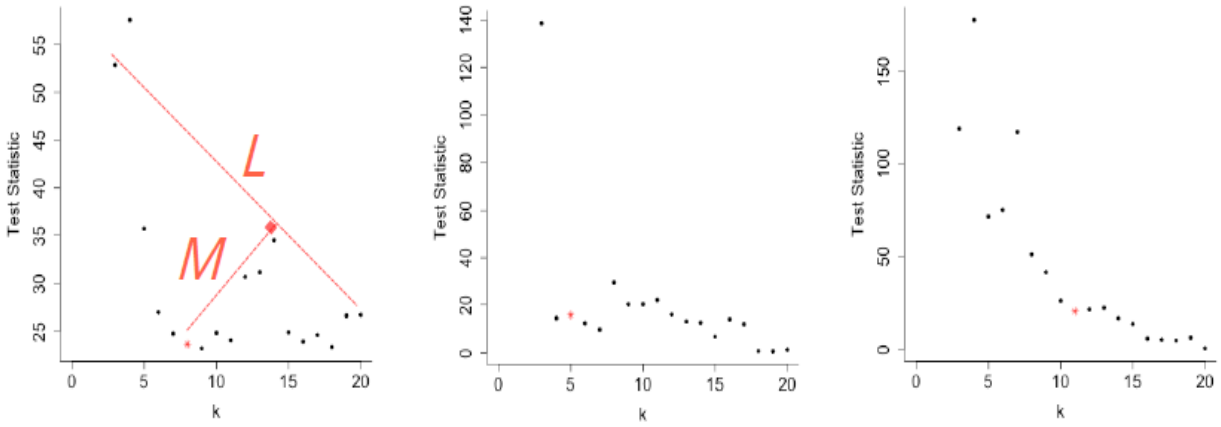

Рис.2.13. Определения оптимального числа кластеров через t-критерий.

Далее результирующая маска с афтофлуоресцентными 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 обастей 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.

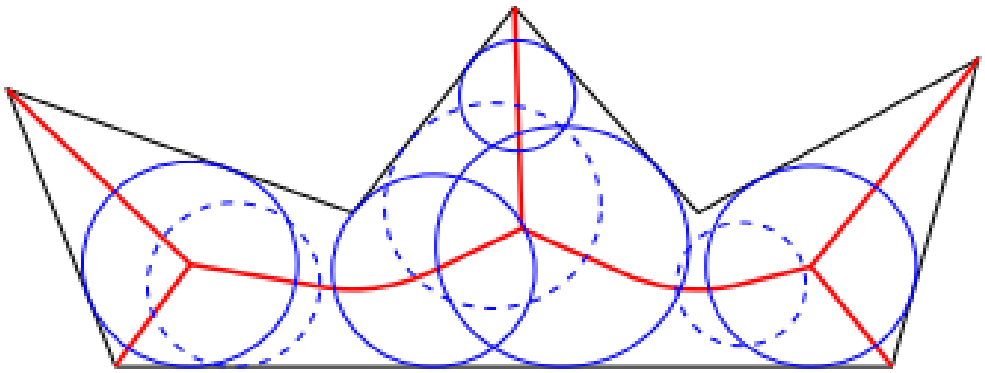

Рис.2.14. Скелет - красная кривая, вписанные круги фигуры - синие.

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]

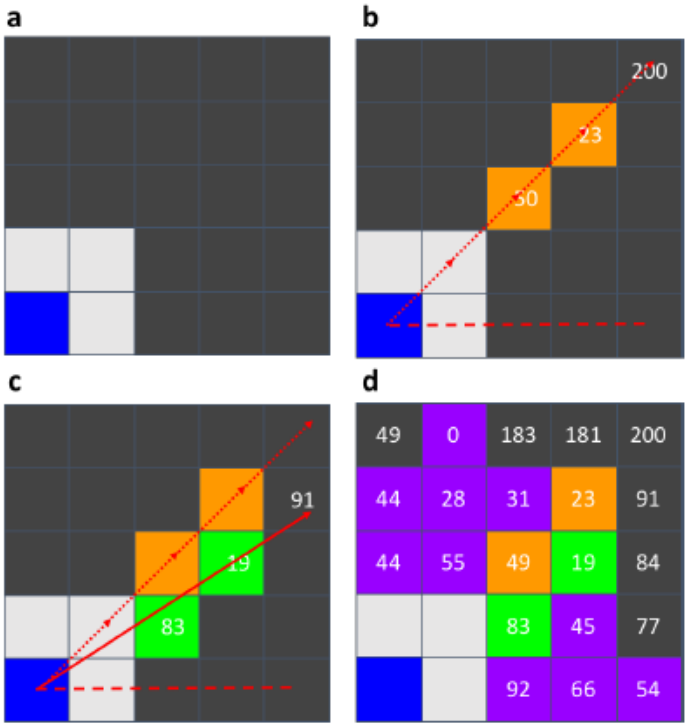

Рис.2.15. Схема иллюстрирующая шаги алгоритма расширения объектов.

На рисунке 2.15 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, обознаены зеленым и фиолетовыми цветами. Все цветные пиксели теперь образуют новый расширенный объект, который изначально частично состоял из 4 пикселей (левый верхний рисунок a).[5]

2.2.5. Обзор алгоритма

У алгоритма AFid существует три вариации:[5]

1. Алгоритм считает автофлуоресцентными те области, межканальный коэффициент корреляции Пирсона значений интенсивностей которых больше некоторого изначально заданного значения.
2. Как было указано в 2.2.3 - вычисляются характеристики для регионов, формируются столбцы со значениями характеристик и далее происхоит кластеризция наборов вычисленных характеристик для каждого региона. Отбирается один кластер у которого среднее значения коэффициента корреляции наибольшее. Все регионы соотвествующие этому кластеру считаются автофлуоресцентными. В данной вариации алгоритма нужно задать правильное количество кластеров.
3. В последнем варианте алгоритма число кластеров выбирается автоматически. Графически это было показано на рис. 2.13

Далее представлены псевдокоды алгоритмов обнаружения автофлуоресценции AFid.[5]

Псевдокод алгоритма автоматического определения оптимального числа кластеров.

Input: table - таблица значений характеристик для каждого региона

Output: resReg - список регонов соответствующие автофлуоресценции.

```
1. Stats=[] //список значений статистик для определеня числа кластеров;
2. for (k) ∈ range(2, 20) do
3.     //кластеризуем наборы характеристик для каждого региона
         kmean=KMeans(table);
4.     //находим первые два кластера с макимальным средним значением
         коорреляций   crMax,crSecMax = findMax(kmean,table);
5.     //отбираем значения корреляций соответствующие двум кластерам
         на прошлом шаге  corrVals1,corrVals2 = findCor(crMax,crSecMax);
6.     //находим значение статистики и добавляем в список
         Stats.append(ttest(corrVals1, corrVals2));
7. kBest = k_best(Stats) //Определяем оптимальное число кластеров;
8. resRegF=findRegAuto(table) //Находим регионы автофлуоресцении;
9. return resReg;
```

Псевдокод алгоритма детекции автофлуоресценции

Input: ch1 - Первый канал,ch2 - второй канал, resmask - маска пересечения, k - число кластеров, kAuto - если не ноль, то определить оптимальное число кластеров

Output: resReg - список регонов соответствующие автофлуоресценции.

```
1.  im1PixStr = regionprops(im1) //поиск регионов в 1 канале;
2.  im2PixStr = regionprops(im2) //поиск регионов в 1 канале;
3.  minArea = 20;
4.  delMin(minArea) //удалить регионы площади меньше minArea;
5.  corr=corrcoef(im1Val,im2lVals)//подсчет коэффициентов корреляций ;
6.  if k > 1 then
7.      std1Vals = std(im1PixStr) //столбец стандартных отклонений;
8.      std2Vals = std(im2PixStr);
9.      kurt1Vals = kurt(im1PixStr) //столбец коэффициентов эксцесса;
10.     kurt2Vals = kurt(im2PixStr);
11.     //нормализуем значение полученных столбцов;
12.     corrNorm = arctanh(corr) / std(arctanh(corr));
13.     std1ValsNorm = log(std1Vals) / std(arctanh(std1Vals));
14.     std2ValsNorm = log(std2Vals) / std(arctanh(std2Vals));
15.     kurt1ValsNorm = log(kurt1Vals) / std(arctanh(kurt1Vals));
16.     kurt2ValsNorm = log(kurt2Vals) / std(arctanh(kurt2Vals));
17.     //Формируем таблицу столбцов нормированных характеристик для
          регионов;
18.     table=tb(corrNorm,std1ValsNorm...kurt2ValsNorm);
19.     if kAuto > 0 then
20.         //Псевдокод алгоритма представлен тут 9;
21.         resRegF=findRegAuto(table) //Находим регионы
              автофлуоресцении;
22.     if kAuto = 1 then
23.         kmean=KMeans(table)//кластеризуем наборы характеристик;
24.         crMax = findMax(kmean,table)//кластер с макимальной
              коорреляций;
25.         resReg=findReg(crMax)//отбираем соотвествующие регионы ;
26. return resReg;
```

Псевдокод алгоритма поиска точек расширения автофлуоресценции

Input: ch1 - Первый канал,ch2 - второй канал, maskAF - маска автофлуоресценции

Output: glow1,glow2 - маски точек расширения для 1 и 2 каналов.

```
1. skel_af = sk(maskAF)//получаем скелет маски автофлуоресценций;
2. end_nodes = findEnd(skel_af)//находим конечные точки скелета, эти
   точки нужны для определения остальных точек расширения;
3. exp_points=findExpPoints(skel_af)//находим остальные точки через
   каждые trace_count пикселей, эта константа задается заранее;
4. //удаляем шум путем размытия для нахождения расширений;
5. im1_blure=GaussianBlur(im1);
6. im2_blure=GaussianBlur(im2);
7. //находим маски расширения для каждого канала используя найденные
   точки раньше, расширения продолжается до тех пор пока значение
   интенсивности пикселя не начнет увеличиваться;
8. glow1, glow2 = findExp(im1_blure, im2_blure, exp_points);
9. return glow1, glow2;
```

На рисунке 2.16 продемонстрированы шаги алгоритма: сверху(a) показаны шаги для детекции автофлуоресценции, в левом нижнем углу(c) шаги для детекции точек расширения найденных на предыдущем шаге автофлуоресценций. [5]

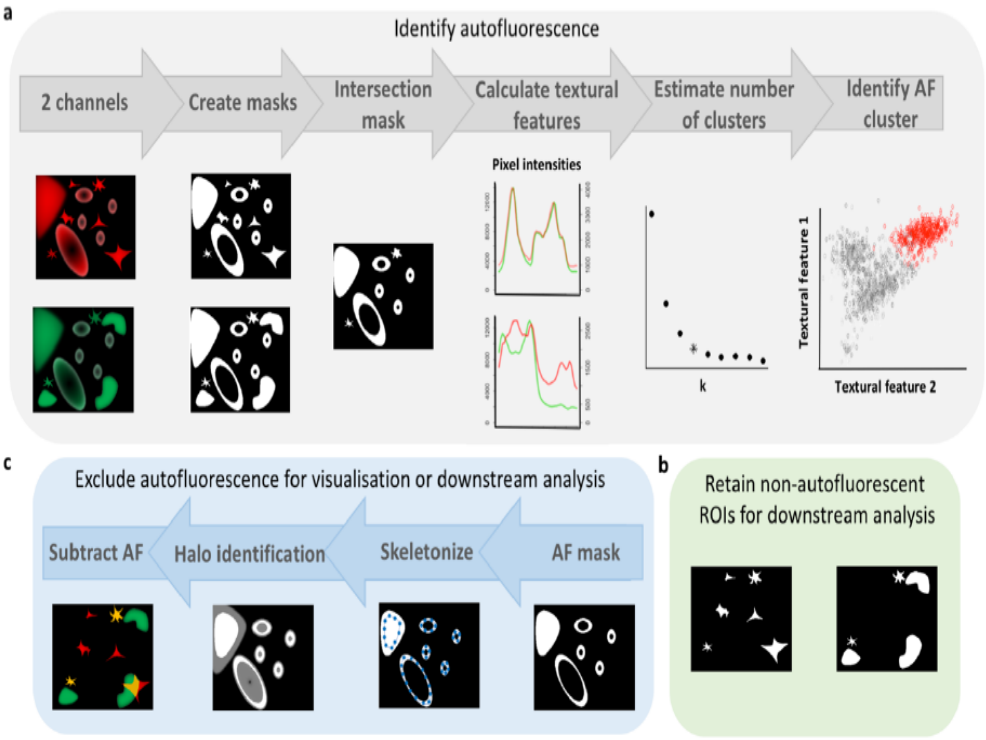

Рис.2.16. Иллюстрация шагов алгоритма AFid.

ГЛАВА 3. ПРИМЕНЕНИЕ МЕТОДОВ

3.1. Модификация сценария ProStack

3.1.1. Описание сценариев

Для выделения комплексов молекул РНК на исходных данных этой работы применяются два сценария:

Первый сценарий - *smooth* отвечает за поворот исходного изображения мозга мушки на угол так, чтобы мозг принял обычное горизонтальное положение. Это делается для дальнейшего правильного определения пространственной локализации кластеров генов. Также в этом сценарии происходит обрезка изображения, чтобы в результат попадал только мозг, без пустого окружающего фона.

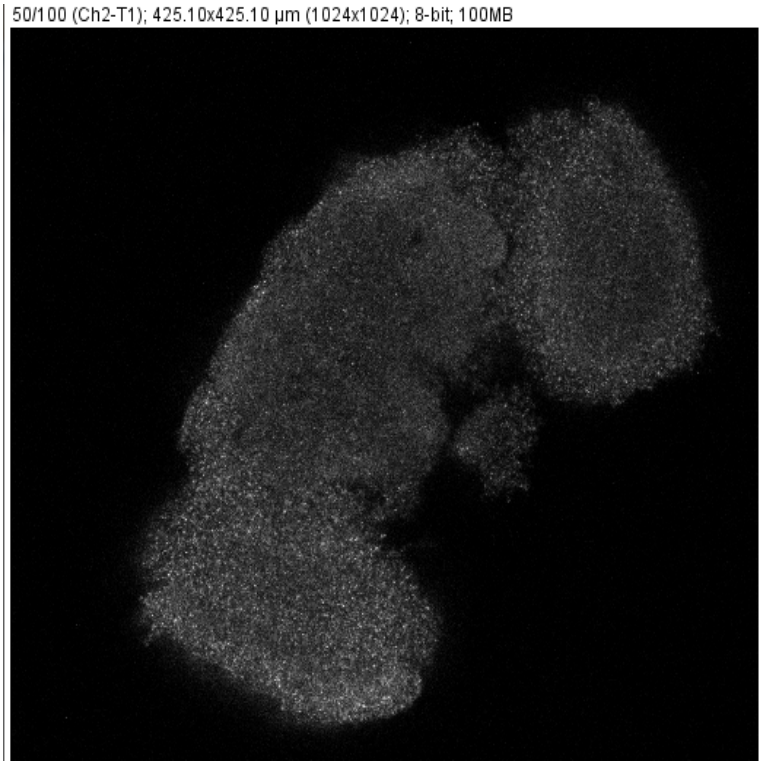

Рис.3.1. Серединный срез 1-го канала изображения мушки дикой породы Sz-139 M5.

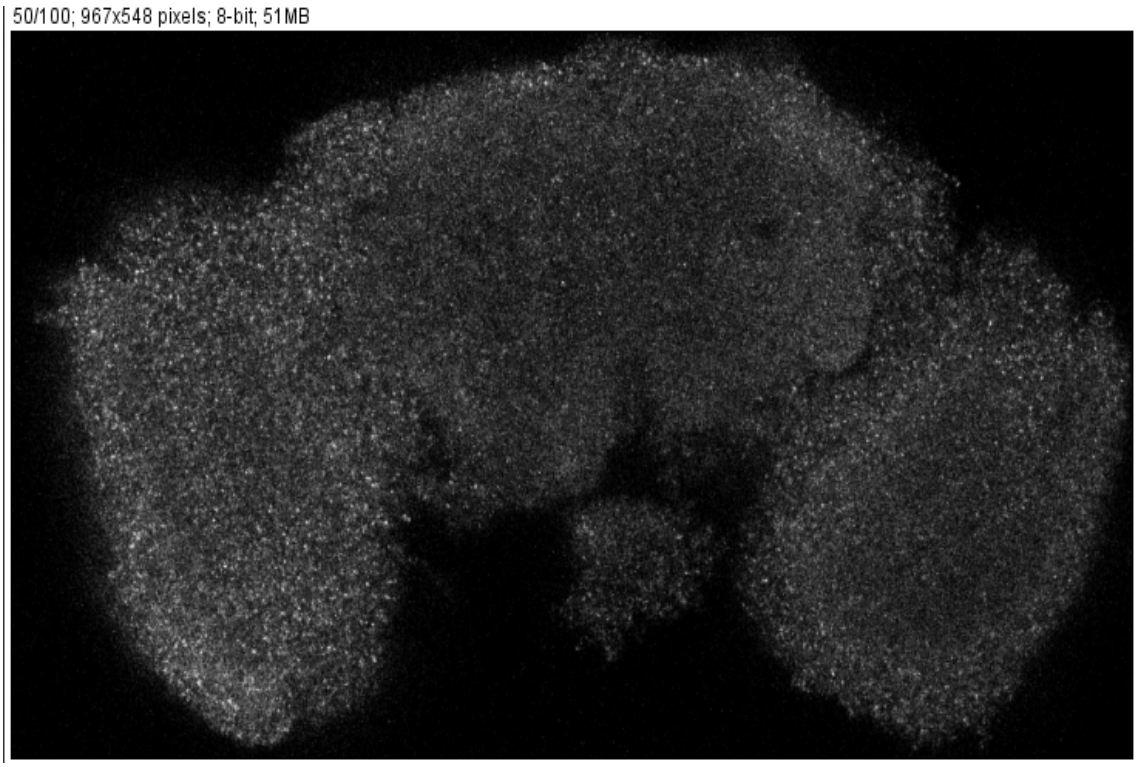

Рис.3.2. Результат обработки сценария *smooth* серединного среза 1-го канала изображения мушки дикой породы Sz-139 M5.

Второй сценарий - *doots* применяется к результату полученному через предыдущий сценарий (*smooth*). То есть получает на вход повернутый и обрезанный участок мозга мушки. Данный сценарий занимается выделением на изображении комплексов молекул РНК и получением количественных данный экспрессии генов: для каждого найденного участка комплекса молекул РНК сохраняется информация о значении интенсивности пикселей, максимальное значение интенсивности, координаты, номер слоя и др.

Маска комплексов молекул РНК получаемая после применения к изображению на рис. 3.2:

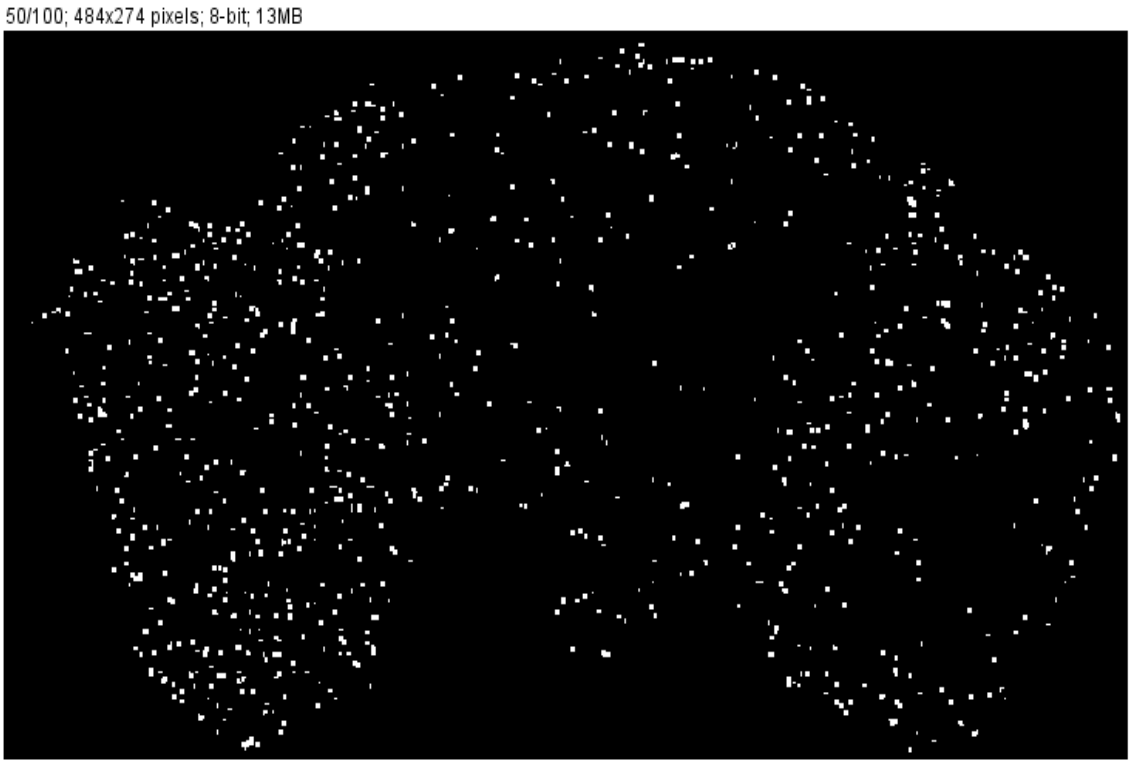

Рис.3.3. Результат обработки сценария *doots* серединного среза 1-го канала изображения мушки дикой породы Sz-139 M5.

3.1.2. Модификация сценариев

Для сценария *doots* внесена следующая модификация:

Для начала рассмотрим участок сценария *doots* в котором происходит удаления фона изображения на рис. 3.4. Крайний правый блок (обозначен как *vaff*) удаляет фон, на входе принимает результат работы *smooth* (обозначен как *-sch2-eve*) и результат медианного фильтра по оцененному фону для *-sch2-eve* и находит разность этих входных изображений (из исходного вычитается фон).

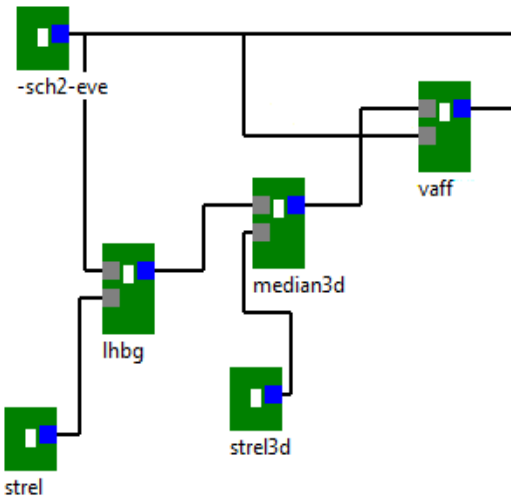

Рис.3.4. Участок doots для удаления фона.

Если применять такую же процедуру удаления фона повернутого и обрезанного участка изображения после применения инструмента удаления автофлуоресценции AFid, то результат будет отличен от ожидаемого. Дело в том, что автофлуоресцентные участки заменяются нулевым значениями интенсивности, а значит эти пустые участки будут вносить иной вклад для оценки фона изображения мозга мушки. Результат показан ни рис.3.5.

Рис.3.5. Результат вычитания фона для 25-го слоя изображения модельной мушки M4 после удаления автофлуоресцентных объектов.

Было принято решение вычитать из обрезанного и повернутого изображения после удаления автофлуоресцентных объектов (обозначение *sch2-eve after AFid*) фон изображения до удаления автофлуоресценции (*-sch2-eve*). Измененная процедура удаления фона показана на рис.

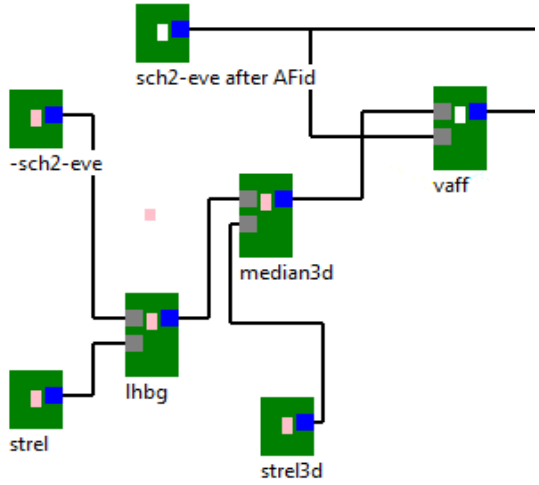

Рис.3.6. Модификация участка doots для удаления фона.

Результат такой модификации показан на рисунке 3.7. Можно заметить, чтобы остаточный фон для некоторых областей, в отличие от результата на рис. 3.5 - исчез, а светящиеся точки остались. Значит удаление фона произведено верно.

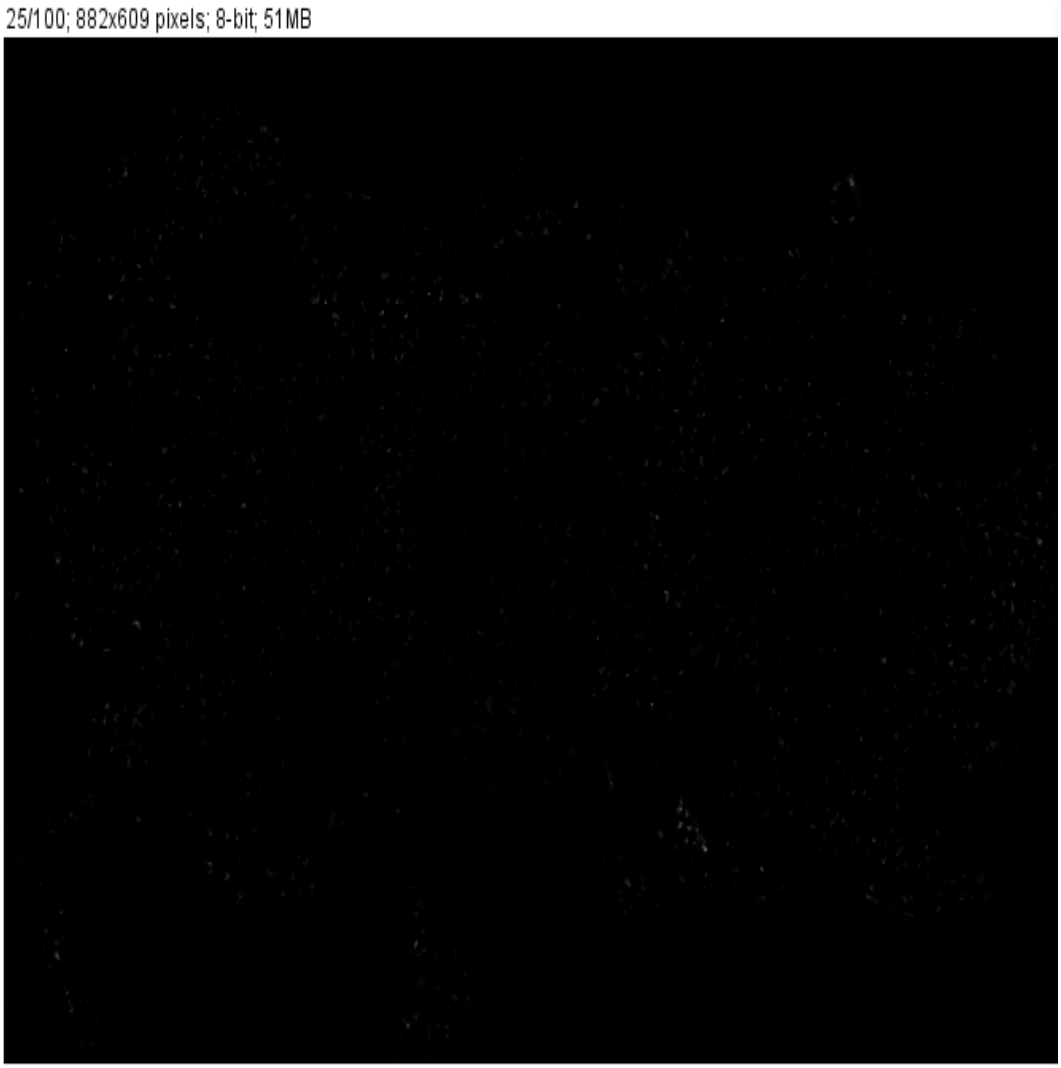

Рис.3.7. Результат вычитания фона для 25-го слоя изображения модельной мушки M4 после удаления автофлуоресцентных объектов с учётом модификации.

После модификации сценария *doots*, также потребовались изменения для сценария *smooth*. Дело в том, что размеры результирующего изображения модифицированного сценария *smooth* (обозначали как *sch2-eve after AFid*) и оцененный фон для -*sch2-eve* - результат отработки *smooth* для оригинального изображения (до применения AFid) для вычитания должны иметь одинаковые размеры. После применения инструмента удаления автофлуоресценции, некоторые области мозга удалялись. Это повлияло на результат обрезки сценария *smooth*. То есть в этом случае изображение получалось немного меньше (разница в несколько десятков пикселей).

Чтобы размеры изображений совпадали, была произведена модификация сценария *smooth* для оригинального изображения. Теперь в сценарии для изображения после удаления автофлуоресцентных объектов сохраняется параметры для обрезания окружающего пустого фона. И эта сохраненная информация исполь-

зуется в сценарии *smooth* для оригинального изображения (до применения AFid). Таким образом, обрезание изображений получается одинаковым и размерности входных данных в блоке удаления фона совпадают.

3.2. Алгоритм выделения границ Canny

Для вычисления количественных данных экспресси генов по изображениям мозга плодовой мушки в сценарий пакета ProStack необходимо добавить алгоритм для выделения границ объектов. В данной работе объекты представляют собой комплексы молекул РНК.

В качестве алгоритма поиска границ был выбран Canny. Это решение связано с низкой вприимчивости к шуму алгоритма и реализацией этого метода в большом количестве пакетов для обработки изображений.

Для применения алгоритма была использована библиотека алгоритмов компьютерного зрения OpenCV. [6]

Для обнаружения границ в исходных данных работы подбирались параметры алгоритма. Так, например, для изображения M7 выделение границ работало хуже с параметрами, которые применялись к остальным изображениям. Для M7 размерность ядра Гауссового размытия было выбрано 4x4 в отличие от 3x3 которое использовалось для остальных изображений.

Параметры алгоритма и шаги выполнения подробно описаны в пункте 2.1. Параметр нижней и верхней границы для утончения границ из шага 4 были выбраны как 30 и 60 соответственно.

На рисунках 3.8 и 3.9 представлен пример применения алгоритма обнаружения границ к исходным данным данной работы.

Рис.3.8. Серединный срез результата удаления фона изображения M7 мушки дрозофиллы.

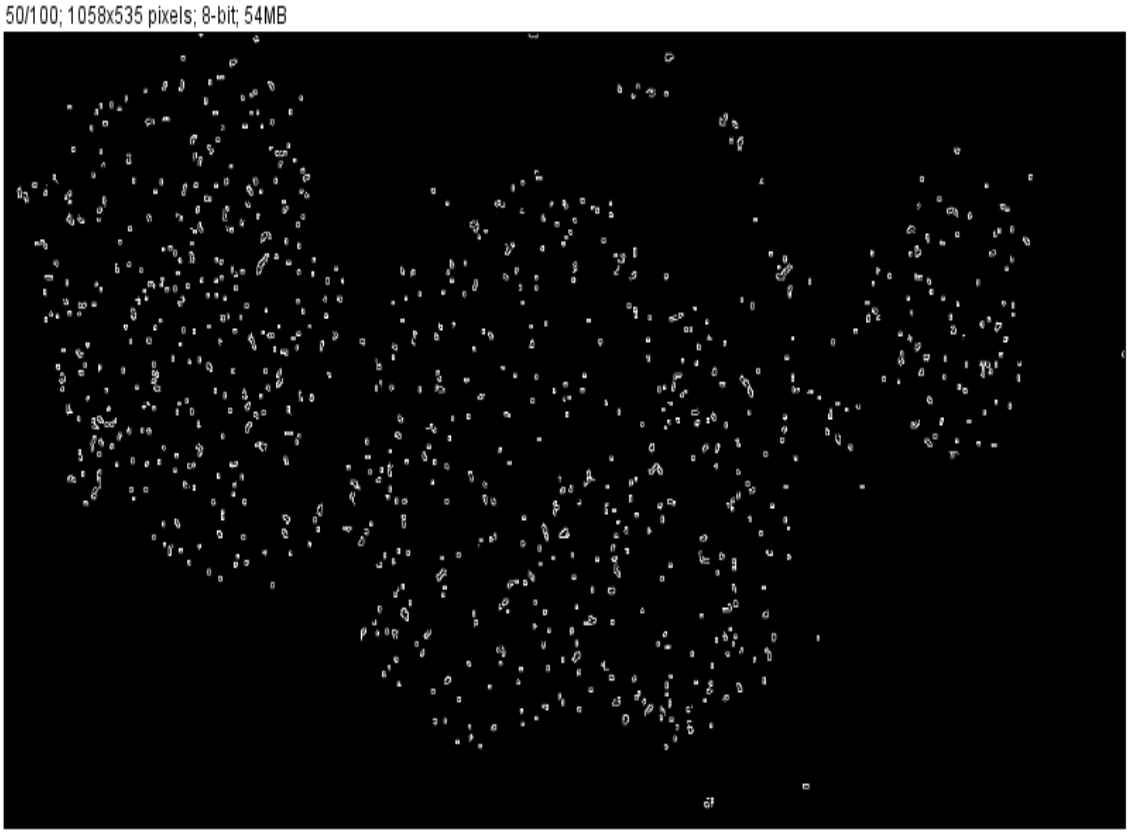

Рис.3.9. Серединный срез результата выделения границ объектов после удаления шума изображения M7 мушки дрозофиллы.

3.3. Модификация AFid

3.3.1. 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.

3.3.2. 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 3.10)

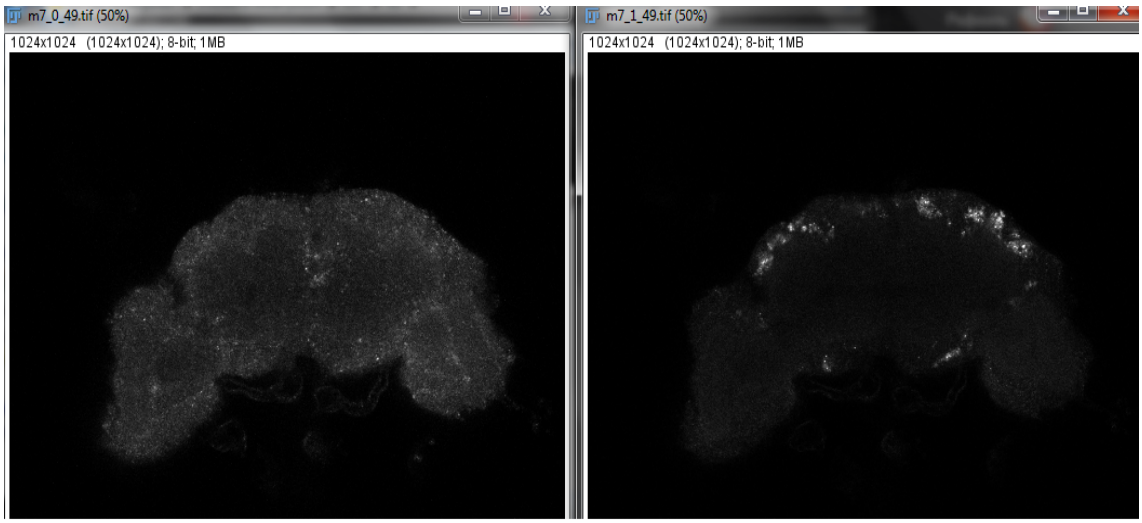

Рис.3.10. Серединный срез многослойного трехмерного двухканального изображения R338 M7 мушки дрозофиллы.

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. 3.11)

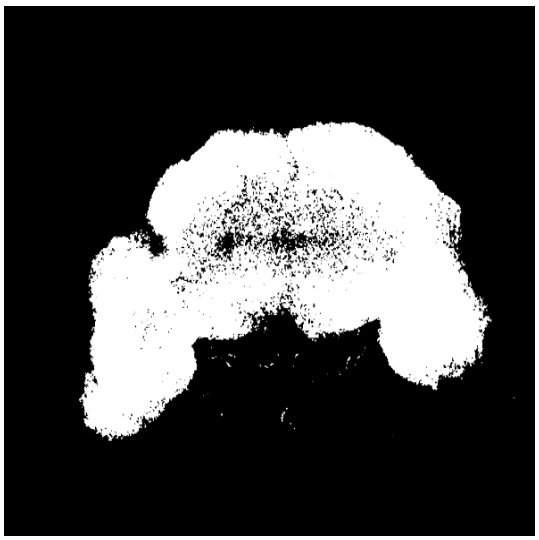

Рис.3.11. Маска пересечения до удаление фона.

Было принято решение реазиловать предобработку входных изображений - удалить фон мешающий иентификации областей автофлуоресценции и реальных сигналов. Удаление фона имеет следующую процедуру:
Расчитывается некоторое пороговое значение d определемое как
$d = meanvalue + coeff * std$
Тут $meanvalue$ - среднее значение пикслей изображения, $coeff$ - эмперически подобранный коэффициент равный 0.8, std - стандартное отклонений значений пикселей изображения.

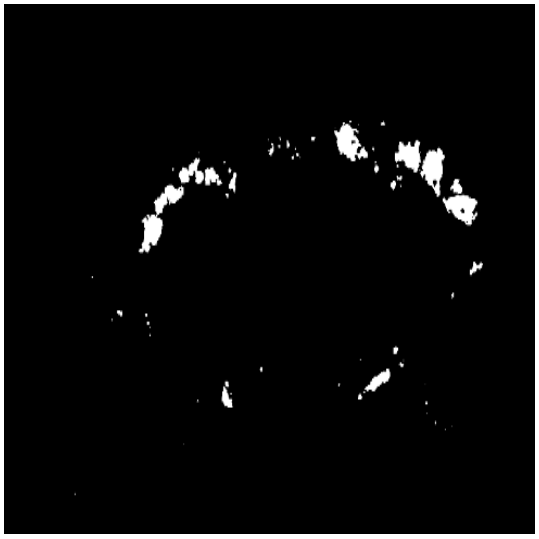

Рис.3.12. Маска пересечения после удаления фона.

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 обеих каналах а не только в первом.(см. рис. 3.12)

3.3.3. Настройка параметров

Для применения алгоритма AFid к исходным данным необходимо эмперически подобрать значения параметров, чтобы алгоритм работал как минимум адекватно (не удалял все объекты и находил хотя бы какие-то).

С помощью отладки программы были выявлены некоторые нижние и верхние пороги для параметров применительно к исходным данным:

1. *Минимальная и максимальная площадь объектов на изображении*. После нахождения связных областей и свойств для каждого канала изображения, необходимо удалить очень маленькие по площади области. Так, при вычислении межканального коэффициента корреляций Пирсона значений пикселей для каждой области - возникали коэффициенты корреляций равные или очень близкие к 1 или -1, или вовсе появлялись неопределённые значения. Это происходило потому, что области с малой площадью (около 3-4 пикселей) представляли из себя шум, где значение всех пикселей а иногда почти всех - совпадали. Отсюда при вычислении коэффициента корреляций получались указанные выше значения. Также не разумно брать очень большие области, так как иначе при хорошом разделении участков мозга для первой вариации алгоритма с межканальной корреляцией (см.2.2.5) в качестве объектов автофлуореценции могут завхватить слишком большие участки мозга. Поэтому нужен верхний порог на площади областей.

   Путём сравнения результатов были выбраны следующие пороги: 7 и 50000 - минимальный и максимальный порог для площади связной области соотвественно.
2. *Значения порогового коэффициента корреляции Пирсона для первой вариации*. Изначально стандартным значением для алгоритма было выбрано 0.6, после попытке применить такой параметр к данным этой работы - маски автофлуоресецнтных объектов получались пустыми. Эмперически было подобрано значение 0.3, при меньших значениях появлялись артефакты - громоздкие участи мозга.

3.4. Применение AFid

Алгоритм AFid применялся к исходным данным описанным в 1.1 (5 изображений модельной мушки и 4 дикой). Для каждого изображения применялись все три вариации алгоритма (вариации описаны в 2.2.5) - чтобы сравнить их и выбрать наилучший судя по полученным результатам. (добавить примеры каждого варианта работы)

После обработки инструментом обнаружения и удаления автофлуоресценции, изображения обрабатывались сценарием пакета Prostack для выделения комплексов молекул РНК и получени количественных данных (интенсивность значений пикселей, координаты выделенных объектов).

3.5. Проверка статистической гипотезы

Инструмент AFid идентифицирует и удаляет автофлуоресцентные объекты. Следовательно, гипотетически, после применения алгоритма паразитное свечение из одного канала в другом должно исчезнуть. Нужно проверить данное предположение. Было решено использовать 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 првоерки 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.

Принцип работы критерия Вилконсона в том, что для каждого изображения в двух каналах вычисляются изменения значений среднеквадратичных отклонений значений интенсивностей пикселей с учетом знака. Далее, изменения сортируются без учета знака и каждому значению в отсортированом ряду присваивается ранг. Подсчитывают суммы рангов для отрицательных изменений и положительных. На основе полученных сумм формируется статистика которую сравнивают с табличным значением квантиля уровня $1 - \frac{\alpha}{2}$ распределения Лапласса. Где α - уровень значимости.

3.6. 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.
(ДОПОЛНИТЬ ПОСЛЕ ВСЕХ РЕЗУЛЬТАТОВ)

3.7. Название параграфа

ГЛАВА 4. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ

4.1. Результат првоерки статистической гипотезы

Критерий Вилконсона показал следующие результаты:

Пять изображений **модельной** мушки: согласно критерию изменение среднеквадратичных отклонений значений интенсивностей пикселей до и после применения инструмента удаления автофлуоресценции AFid статистически значимо при $\alpha = 0.5$ для вариации алгоритма с автоматическим определением оптимального числа кластеров (3 вариант, см. 2.2.5) и вариации с межканальным пороговым значений коэффициентов корреляции Пирсона (1 вариант, см. 2.2.5). А значит можно утверждать, что для двух вариаций алгоритма AFid действительно паразитное свечение удаляется.

Для вариации с числом кластеров равным шести (2 вариант, см. 2.2.5), согласно критерию гипотеза принимается только для первого канала. Для изображений M4, M6, M7 второго канала изменений среднеквадратичных отклонений нет. А значит свечение не удалилось.

Четыре изображения мушки **дикой породы** Sz-139: для вариации с числом кластеров равным шести в изменение среднеквадратичных отклонений значений интенсивностей пикселей до и после применения AFid статистически значимой разницы также нет.

Вариации алгоритма с автоматическим определением оптимального числа кластеров и с межканальным пороговым значений коэффициентов корреляции Пирсона согласно критерию Вилконсона удаляют паразитное свечение при уровне значимости 7%. То есть в отличие от модельной мушки - для дикой породы в изменении среднеквадратичных отклонений статистическая значимость более слабая (на уровне меньше 10%)

α - уровень значимости.

4.2. Результаты кластеризации и фильтрации

4.2.1. Результат фильтрации

Ниже приведены построенные гистограммы для удалени шума изображения модельной мушки M4.

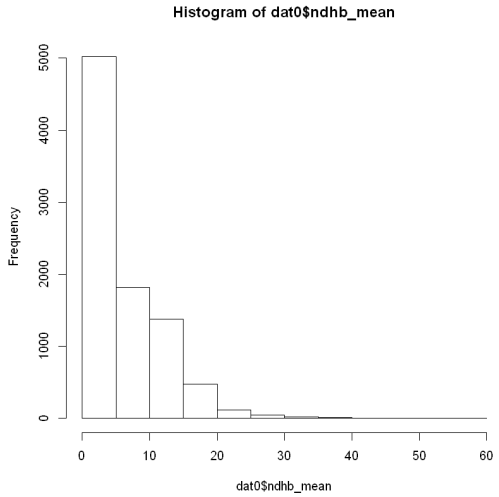

Рис.4.1. Гистограмма частоты встречаемости пикселей по мере увеличения значения интенсивности для изображения M4.

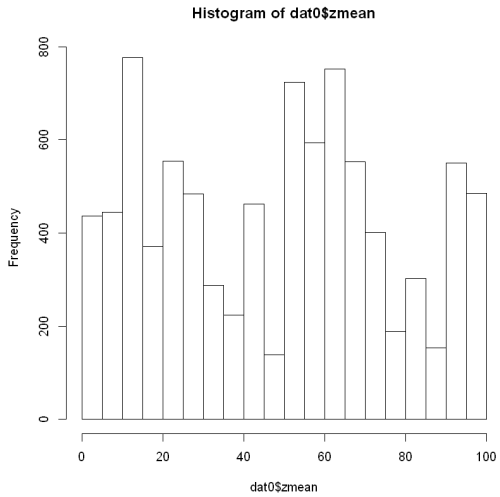

Рис.4.2. Гистограммма показывающая количество комплексов молекул для каждого слоя изображения M4.

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

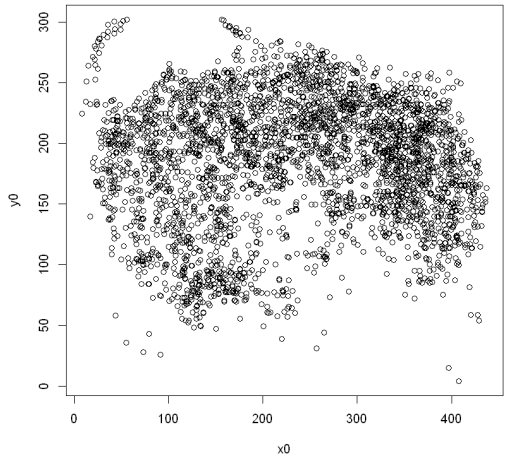

Рис.4.3. Результат фильтрации для изображения мозга модельной мушки M4.

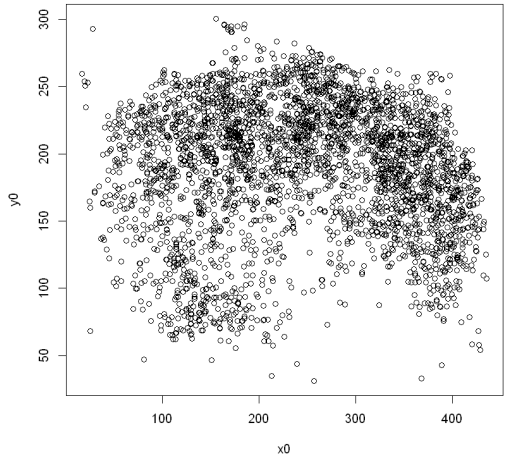

Рис.4.4. Результат фильтрации для изображения мозга модельной мушки M4 после применения инструмента удаления автофлуоресценции AFid.

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.

(ДОПОЛНИТЬ ПОСЛЕ ВСЕХ РЕЗУЛЬТАТОВ)

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 автофлуореценции 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 изображением.
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.