Т.Г. Коновалова, В.М Комашко*

ОБРАБОТКА ДАННЫХ МИКРОЧИПОВЫХ ЭКСПЕРИМЕНТОВ ПРИ ПОМОЩИ ЯЗЫКА «R»

ВВЕДЕНИЕ

В решении тех многих вопросов, что встали перед биологами в постгеномную эру, существуют два основных подхода, предлагаемых функциональной геномикой. Первый подход основан на исследовании нуклеотидных последовательностей, приведший к обнаружению различных мотивов, сайтов и структурных доменов. Второй подход — для изучения функции гена — основан на исследовании модели его экспрессии (экспрессия — продукция гена, коррелирующая с конечным количеством белка, произведенным данным геном). Новым классом биотехнологий для одновременного отслеживания изменения уровня экспрессии десятков тысяч генов в разных экспериментах и под влиянием различных условий стала технология микрочипов.

В данной статье рассматривается подход к решению задачи выделения генов, изменивших с определенной достоверностью уровень своей экспрессии на основе данных из нескольких микрочиповых экспериментов. Эксперименты заранее разбиты на две группы. Над этими данными вначале проводится нормализация, а затем — кластеризация. В данной статье под кластеризацией подразумевается выделение двух кластеров — гены, изменившие и не изменившие свою экспрессию.

Существуют две основные технологии синтеза микрочипов: кДНК микрочипы и синтез *in situ (олигонуклеотидные чипы)* — чипы, состоящие из олигонуклеотидов (цепочек ДНК длины 10–20 нуклеотидов). кДНК микрочипы предоставляют относительный уровень экспрессии каждого отдельного гена, в то время как олигонуклеотидные чипы дают информацию об абсолютном уровне экспрессии.

После сканирования и преобразования изображения в цифры, отражающие уровень экспрессии каждого гена, исследователю предоставляется матрица размера $n \times l$, где n — это количество проведенных эксперимен-

* tiv_@ngs.ru

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 05-08-33448).

тов, будь то сравнение между тканью, пораженной раком, и такой же, но здоровой, или исследования экспрессии генов в ходе клеточного цикла, где каждый столбец представляет собой временную точку; *l* — обозначает количество проб (проба соответствует гену или части гена), причем *l* может варьировать от 100 до десятков тысяч. В такой матрице необходимо найти ключевые гены, экспрессия которых изменилась, для чего было разработано множество методов. Для достоверности результатов и возможности сравнения полученных данных из различных источников, результаты эксперимента до поиска ключевых генов должны быть нормализованы, т. е. приведены к единой шкале. Это, а также разработка единого языка описания, позволяют сделать данные доступными и сопоставимыми.

Во многих случаях целью микрочипового эксперимента является сравнение уровней экспрессии в двух различных образцах. В большинстве случаев один образец рассматривается как контрольный, а второй — как экспериментальный. При этом основной задачей является определение таких генов, экспрессия которых различается при сравнении двух исследуемых образцов. Несмотря на то что на первый взгляд эта проблема не кажется сложной, она становится такой, из-за того что на полученные значения интенсивности влияет множество эффектов и шума, связанных с технологическими ограничениями эксперимента.

Решение задач нормализации и кластеризации данных удобно выполнять в среде программирования языка *R GUI* [R Development Core Team, 2004] с использованием пакетов *Bioconductor* [Ihaka, Gentleman, 1996]. В данной статье описан процесс обработки данных с олигонуклеотидного чипа компании Affimetrix. В эксперименте рассматривалась экспрессия генов в клетках раковых опухолей мозга. Для анализа мы взяли результаты сканирования 12 чипов (по 6 для глио- и олигобластомы, являющихся опухолями вспомогательных тканей мозга).

Данная работа может быть использована как руководство по применению языка R и пакета BioConductor для обработки результатов микрочиповых экспериментов.

Список терминов

Проба — олигонуклеотид из 25 пар оснований, используемый для связывания РНК мишеней.

PM, perfect match — пробы, разработанные так, чтобы идеально совпадать с последовательностью мишени.

MM, mismatch — пробы, имеющие замену в одном нуклеотиде, для учета неспецифического связывания.

affyID — идентификатор набора проб, соответствующий гену или части гена.

CEL файл — файл простого текстового формата, полученный после цифровой обработки цветного изображения микрочипа и содержащий информацию об интенсивностях для PM/MM пар, а также физические координаты каждой клетки на слайде.

1. МЕТОДЫ НОРМАЛИЗАЦИИ

Пакеты используемого в данной работе модуля *Bioconductor* позволяют своим пользователям применять несколько широко используемых алгоритмов, а также комбинировать их с помощью функции *expresso*(). Одним из наиболее быстрых и эффективных является метод *RMA* [1], который включает в себя несколько этапов.

1. Коррекция по фону — *rma*. Так как, в отличие от кДНК микрочипов, олигонуклеотидный чип вообще не содержит фона, то коррекция осуществляется за счет соседних проб для учета неспецифического связывания и оптического шума. Интенсивности *PM* корректируются в глобальной модели для распределения интенсивностей проб. В данной модели предусматривается рассмотрение кривой эмпирического распределения интенсивностей проб. В частности, наблюдаемые *PM (Perfect Match)* пробы моделируются как сумма нормальной шумовой компоненты N (нормальной со средним μ и дисперсией σ^2) и экспоненциальной сигнальной компоненты S (экспоненциальная со средним α). Для избежания вероятности отрицательных значений нормаль округляется до нуля. Пусть O — это наблюдаемая интенсивность, тогда корректировка будет выполнена по следующей формуле:

$$E(s \mid O = o) = a + b \frac{\phi\left(\frac{a}{b}\right) - \phi\left(\frac{o-a}{b}\right)}{\Phi\left(\frac{a}{b}\right) + \Phi\left(\frac{o-a}{b}\right) - 1},$$

где $a = s - \mu - \sigma^2\alpha$ и $b = \alpha$. Здесь ϕ — плотность стандартного нормального распределения, а Φ — это распределение функций.

Необходимо заметить, что в этом методе для коррекции фона значения *MM* в расчет не принимаются.

2. В качестве метода нормализации был выбран метод квантилей [2]. Цель данного метода состоит в том, чтобы сделать распределение интенсивностей проб для каждого чипа одинаковым для всех остальных чипов. Этот метод мотивирован известным в статистике квантиль-графиком, показывающим, что распределение двух векторов данных одинаково, если график представляет собой прямую диагональную линию, и распределение разное, если график имеет другой вид. Увеличим количество векторов до n: если все n векторов имеют одинаковое распределение, то построение квантилей в размерности n дает прямую линию вдоль диагонали, заданную единичным вектором $\left(\frac{1}{\sqrt{n}},...,\frac{1}{\sqrt{n}}\right)$. Это предположение можно сделать для набора данных, имеющих одинаковое распределение, если спроектировать все точки n-размерного квантильного графика на эту диагональ.

Пусть $q_k = (q_{k1},...,q_{kn})$ для $k = \overline{1,p}$ будет вектором k-го квантиля для всех n чипов $q_k = (q_{k1},...,q_{kn})$ и $d = \left(\frac{1}{\sqrt{n}},...,\frac{1}{\sqrt{n}}\right)$ будет единичной диагональю. Чтобы трансформировать квантили так, чтобы они лежали на диагонали, рассмотрим проекцию q на d:

$$proj_d q_k = \left(\frac{1}{n}\sum_{j=1}^{n} q_{kj},...,\frac{1}{n}\sum_{j=1}^{n} q_{kj}\right).$$

Это подразумевает, что мы можем приписать каждому чипу одинаковое распределение, взяв средний квантиль и заместив им значения данных в исходных данных. Это оправдывается следующим алгоритмом.

1. Данные n чипов размера p представляют собой матрицу X размерности $p \times n$, где каждый чип — это столбец в данной матрице.
2. Сортируется каждый столбец матрицы X для получения X_{sort}.
3. Находится среднее значение по рядам матрицы X_{sort} и это значение присваивается каждому элементу в ряду для получения X'_{sort}.
4. Матрица с нормализованными значениями получается перестановкой столбцов матрицы X'_{sort} так, чтобы порядок был таким же, как и вначале.

Одной из возможных проблем этого метода может быть принуждение всех квантилей к равенству. Это может быть наиболее проблематично в наибольшей степени в хвостах распределения, где может случиться так, что

все пробы имеют одинаковое значение для всех чипов. Однако на практике, так как уровень экспрессии вычисляется с использованием всех величин множества проб, этот возможный недостаток не становится реальной проблемой.

Согласно методу расчета коррекции по фону, при коррекции на значения *PM*, значения *MM (MisMatch)* не учитываются вообще, хотя, например, у Affymetrix (2002) при расчете *PM* идеальное значение *MM* отнимается от значения *PM*, при этом *MM* всегда меньше *PM*.

3. Для вычисления меры экспрессии был использован метод *medianpolish*, также предложенный в [1]. Линейная мультичиповая модель состоит в корректировке данных из каждого набора проб. В частности, для набора k с пробами $i = 1,..,I_k$ и данными из $j = 1,...,J$ чипов строится следующая модель:

$$\log_2(PM_{ij}^{(k)}) = \alpha_i^{(k)} + \beta_j^{(k)} + \varepsilon_{ij}^{(k)},$$

где $\alpha_i^{(k)}$ — эффект пробы и $\beta_j^{(k)}$ — логарифм по основанию два величины экспрессии. Алгоритм *medianpolish* нужен для устойчивого прилаживания модели.

Таким образом, полный набор методов для обработки экспериментальных данных включает в себя несколько этапов: 1) коррекция по фону; 2) нормализация, т. е. приведение данных к единой шкале; 3) коррекция значений *PM*; 4) вычисление значения экспрессии.

2. МЕТОДЫ ВЫДЕЛЕНИЯ ГЕНОВ С ДИФФЕРЕНЦИАЛЬНОЙ ЭКСПРЕССИЕЙ

Задача идентификации таких генов разбивается на две части:

1) выбор статистики, которая ранжирует гены, для того чтобы доказать дифференциальную экспрессию — от самого строгого доказательства до самого слабого;

2) выбор критического значения статистики такого, что все значения больше этого критического будут считаться значимыми. Обычно при исследованиях выделяют около 100 генов, для которых возможно показать, что их экспрессия изменилась в сравнении между двумя условиями.

Метод выбора генов может быть характеризован в терминах положительной предсказанной величины (PPV), отрицательной предсказанной величины (NPV), специфичности и чувствительности. В общем, для любой

диагностики или метода классификации можно сравнить полученые результаты с экспериментальными. В ситуации, когда есть возможность принять всего лишь одно решение из двух, например, изменился или не изменился уровень экспрессии, результат всегда можно разделить на четыре категории: уровень действительно изменился и методом сообщен как изменившийся (true positives,TP); уровень не изменился, но метод сообщил о его изменении (false positives, FP, ошибка перепредсказания); действительно изменился, но метод сообщил как о неизменившемся (false negatives, FN, ошибка недопредсказания); действительно не изменился и методом сообщен как не изменившийся (true negatives,TN). Основываясь на этих категориях, можно выделить четыре количественных критерия: PPV, NPV, специфичность и чувствительность (рис.1).

Оценки этих значений обычно варьируют от 0 до 1, иногда их выражают в процентах. Наилучший метод характеризуется отсутствием ошибок недопредсказания и перепредсказания.

	Измененный	Не измененный	
Измененный (метод)	TP	FP	PPV= $\frac{TP}{TP+FP}$
Не измененный (метод)	FN	TN	NPV= $\frac{TN}{TN+FN}$
	Чувствительность $\frac{TP}{TP+FN}$	Специфичность $\frac{TN}{TN+FP}$	

Рис.1. Определение положительной и отрицательной предсказанных величин, специфичности и чувствительности. Для наилучшего метода PPV = NPV = специфичность = чувствительность = 1

Тестирование гипотез, коррекция для множественных сравнений

Другим подходом для выбора генов является использование одномерных статистических тестов. Пусть логарифмы отношений экспрессии будут принадлежать некоторому распределению.

Для данного порога и данного распределения уровень значимости, или величина *p,* — это вероятность того, что измеренная величина случайным

образом будет принадлежать заштрихованной области. Идея заключается в том, что ген, чье значение логарифма отношений попадает в некую область, находится далеко от среднего логарифма отношений уровней экспрессии и будет, таким образом, отнесен к дифференциально экспрессирующимся (экспрессия в данном случае будет повышена). Однако измеренный логарифм отношений может быть таким благодаря внешним факторам, например шуму. Вероятность этого обозначается величиной *p*. В этом случае отнесение гена к эспрессирующимся дифференциально будет ошибкой первого рода и величина *p* — это вероятность сделать такую ошибку.

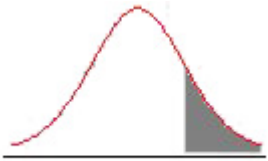


Рис. 2. Схематизированное изображение нормального распределения

Была принята нулевая гипотеза H_0, состоящая в том, что уровень экспрессии генов не поменялся при сравнении его в двух экспериментах. Соответственно, альтернативная или рабочая гипотеза H_1 предполагает наличие изменения уровня экспрессии, или, формализуя определение:

$$H_0: \mu_1 = \mu_2, \; H_1: \mu_1 \neq \mu_2 \; .$$

Для сравнения экспрессии применялась t-статистика:

$$t_j = \frac{\overline{x_{2j}} - \overline{x_{1j}}}{\sqrt{\frac{s_{1j}^2}{n_1} + \frac{s_{2j}^2}{n_2}}},$$

где $\overline{x_{ij}}$ — это среднее значение уровня экспрессии гена *j* в n_1 контрольных и n_2 исследуемых гибридизациях. В нашем случае $n_1 = n_2 = 3$; s_{1j}^2 и s_{2j}^2 — дисперсии.

Множественное тестирование, коррекция значения p

Вопрос множественной коррекции состоит в том, что необходимо полностью контролировать вероятность сделать ошибку первого рода. Эта вероятность равна вероятности сделать по крайней мере одну такую ошибку, и вычисляется она следующим образом:

$$P = 1 - (1 - p)^G, \quad [1]$$

где G — это количество генов. Данное выражение может быть переписано следующим образом:

$$\alpha_e = 1 - (1 - \alpha_c)^G, \quad [2]$$

где α_e — вероятность ошибки первого рода на уровне эксперимента, а α_c — вероятность ошибки для одного гена для одного сравнения. Задача состоит в вычислении такого уровня α, который нам нужно использовать для отдельных генов (α_c), для того чтобы быть уверенными, что глобальная или экспериментальная ошибка первого рода была меньше или равна выбранной α_e.

1. *Коррекция Шидака.* Из выражения [2] найдем значение α_c:

$$\alpha_c = 1 - \sqrt[G]{1 - \alpha_e},$$

где G — количество генов.

2. Коррекция Бонферрони. Для небольших значений p выражение [2] может быть аппроксимировано двумя первыми членами распределения бинома:

$$\alpha_e = 1 - (1 - \alpha_c)^G = 1 - (1 - G * \alpha_c + ...) \approx G * \alpha_c.$$

Используя эту аппроксимацию, вычисляем экспериментальный уровень α_c:

$$\alpha_e = \alpha_c * G \Rightarrow \frac{\alpha_e}{G}$$

3. Скачкообразная коррекция Холма. На первом этапе выбирается экспериментальный уровень значимости α_e. Гены упорядочиваются в порядке увеличения их значений p.

Сравниваются p-значения каждого гена с порогом, который зависит от позиции гена в упорядоченном списке. Порог вычисляется следующим об-

разом: $\frac{\alpha_e}{G}$ для первого гена, а для второго гена $\frac{\alpha_e}{G-1}$ и так далее. Пусть k будет максимальным i, для которой $p_i < \frac{\alpha_e}{G-i+1}$. Нулевая гипотеза будет отвергнута для всех i, меньших или равных k.

4. Метод отношения ошибочного предсказания. Алгоритм схож с методом скачкообразной коррекции Холма за исключением выбора порогов. Пусть k будет максимальным i, для которой $p_k < \frac{i}{G}\alpha_e$. Нулевая гипотеза будет отвергнута для всех i, меньших или равных k.

5. Перестановка. Как уже говорилось, данные после нормализации представляют собой матрицу, где столбцы обозначают различные эксперименты, а строки — гены. В этом методе на первом этапе производится перестановка столбцов матрицы или, что эквивалентно, меток каждого столбца. На каждой перестановке вычисляются новые значения p и корректируются с использованием метода скачкообразной коррекции Холма. Весь процесс — случайная перестановка меток столбцов и тестирование — повторяется сотни или десятки тысяч раз, что зависит от количества столбцов. В конце концов, величина p для гена i будет вычисляться следующим образом:

p для гена I = число перестановок, для которых $u_i^{(b)} \geq t_i$ */полное число перестановок,* где $u_i^{(b)}$ — это величины, скорректированные методом скачкообразной коррекции Холма.

3. ОБРАБОТКА ДАННЫХ В СРЕДЕ ПРОГРАММИРОВАНИЯ ЯЗЫКА *R*

1. Инсталляция R GUI и пакетов Bioconductor

С сайта `http://www.r-project.org/` скачивается и устанавливается файл rw1090.exe.

Установка пакетов Bioconductor осуществляется командами

```
R> source("http://www.bioconductor.org/getBioC.R")
R> getBioC() # Установка всех пакетов по умолчанию
```

или через меню *Packages/ Install package(s)* from *Bioconductor*.

Инсталляция пакетов производится один раз, но перед началом каждой рабочей сессии пакеты должны быть **загружены** командой library(“имя библиотеки”) или через меню *Packages/Load package*.

2. Подготовка среды

Установка директории, содержащей файлы в качестве рабочей (команда setwd("..") или в меню *File/Change Dir).*

Загрузка библиотек, необходимых для работы.

```
R>library(Biobase) ## содержит неспецифические
компоненты, требуется для работы других пакетов
Bioconductor
R>library(affy) ## обеспечивает считывание и
нормализацию данных
R>library(multtest)## функции для множественного
тестирования
R>library(annotate)## аннотация данных
R>library(hgu95av2)## описание платформы,
использованной экспериментатором
```

3. Загрузка данных

Для загрузки данных в R используется команда:

```
R> Data <- ReadAffy() ##считывает все данные в рабочей
директории
```

Чтобы выбрать часть файлов и избежать в дальнейшем отображения названий образцов как путей к файлам, целесообразно воспользоваться следующей последовательностью действий:

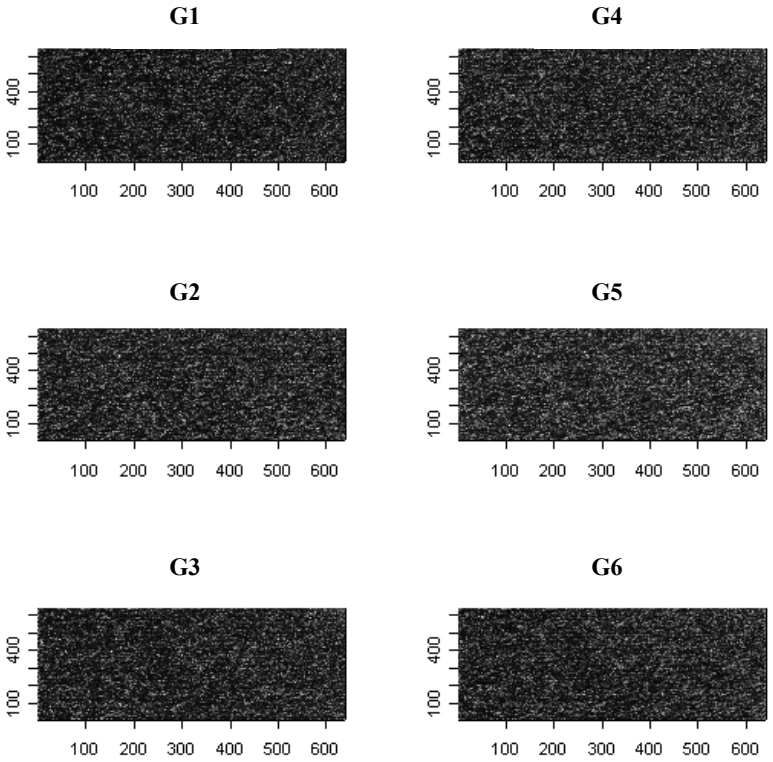
```
R> sample.files<-c("G1.cel","G2.cel","G3.cel",
"G4.cel","G5.cel","G6.cel","O1.cel","O2.cel",
"O3.cel","O4.cel","O5.cel","O6.cel")## первая буква
соответствует типу ткани
R>sample.names<-c("G1","G2","G3","G4","G5","G6","O1","O2",
"O3","O4","O5","O6")
R> data<-ReadAffy(filenames=sample.files)
```

Использование параметра widget (`ReadAffy(widget=TRUE))` позволяет воспользоваться окном браузера для выбора файлов и задания соответствующего описания.

4. Построение диагностических графиков, демонстрирующих необходимость стандартизации экспериментальных данных

Пространственное изображение

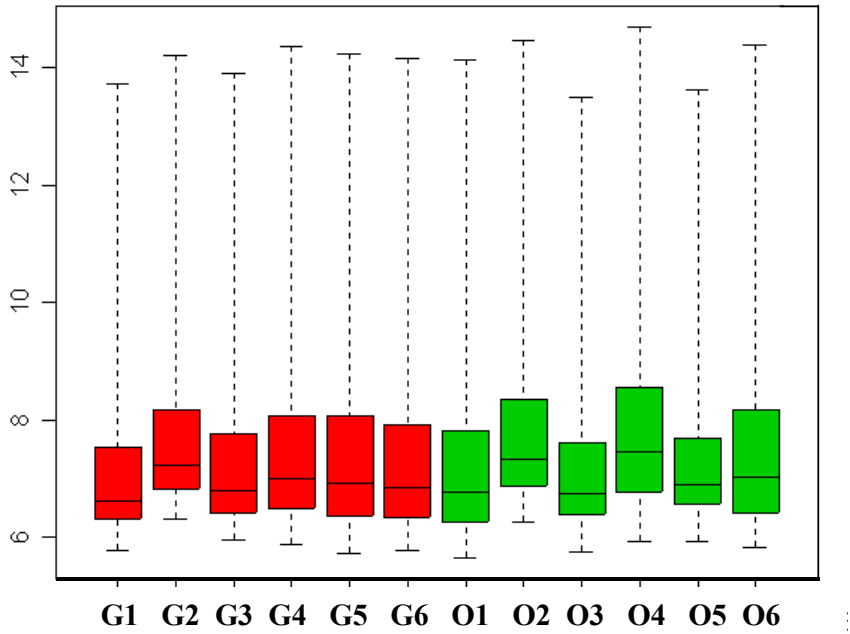
```
R>par(mfcol=c(3,2))#установка параметров графического
окна, (по умолчанию в новом окне будут рисоваться изо-
бражения для каждого файла по очереди)
R>image(data)
```



Данное изображение позволяет визуально оценить качество каждого отдельного чипа. Исследуется наличие царапин, пузырей, яркости и равномерности цвета чипа. Неравномерность цвета говорит о проблемах во время отмывки чипа от негибридизовавшейся ДНК. Как видно из этих изображений: яркость чипа равномерна, нет белых «вспышек», качество хорошее.

Диаграмма размаха

```
R> par(mfcol=c(1,1))
R> boxplot(vih,col=c(2,2,2,2,2,2,3,3,3,3,3,3,3) # col
— указывает цвета
```

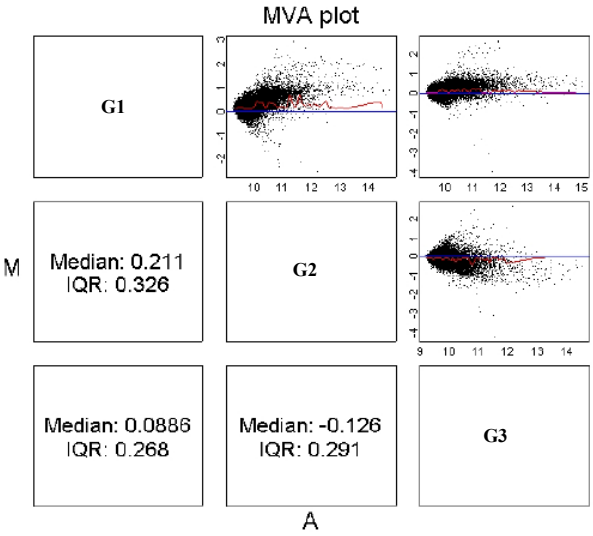


В данной диаграмме сравниваются значения медиан для всех чипов, объем данных для каждого чипа, попадающий в интерквартильное расстояние (50 % данных) и разброс самых крайних точек. На данной диаграмме для шести чипов видно, что значение медианы неодинаково, что говорит о необходимости нормализации данных.

MVA диаграмма

Строится для 1000 генов. Парные сравнения:

```
R> gn<-sample(geneNames(human),1000)
R> pms<-pm(data[,1:3],gn)
R> mva.pairs(pms)
```



Аналогично для остальных сравнений data[,4,6], data[,7,9].

MA диаграмма является наиболее информативной после диаграммы размаха; здесь M — это разница логарифмов экспрессии, а A — это среднее значение логарифма экспрессии. Для любых двух чипов i, j с интенсивностями проб x_{ki} и x_{kj}, где $k = \overline{1, p}$ представляет пробу, вычисляются

$$M_k = \log_2\left(x_{ki} \Big/ x_{kj}\right) \quad A_k = \frac{1}{2}\log_2\left(x_{ki} x_{kj}\right).$$

Диаграмма строится для значений интенсивности PM для всех возможных пар чипов. Для тех случаев, где мы ожидаем небольшое число генов с дифференциальной экспрессией, облако значений должно располагаться вдоль оси $M = 0$, и среднее значение, таким образом, должно быть небольшим. Однако вследствие нежелательных вариаций, связанных с ограничениями технологии, облако значений, как правило, имеет вид, близкий к характерному виду «запятой».

5. Стандартизация данных

Преобразование величин интенсивности проб в уровни экспрессии генов осуществляется следующей последовательностью действий:

- коррекция по фону;
- нормализация;
- коррекция, специфическая для пробы;
- вычисление значения экспрессии.

Можно выбрать один метод, осуществляющий все эти действия, например, метод RMA.

```
R> eset<-rma(data)
```

Также возможно комбинировать разные методы при помощи функции `expresso()`.

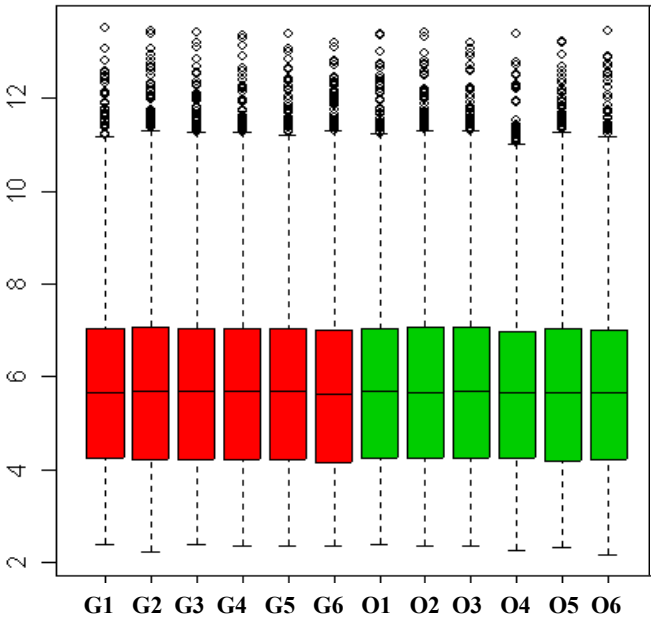
Запись результатов стандартизации в файл, формат .txt:

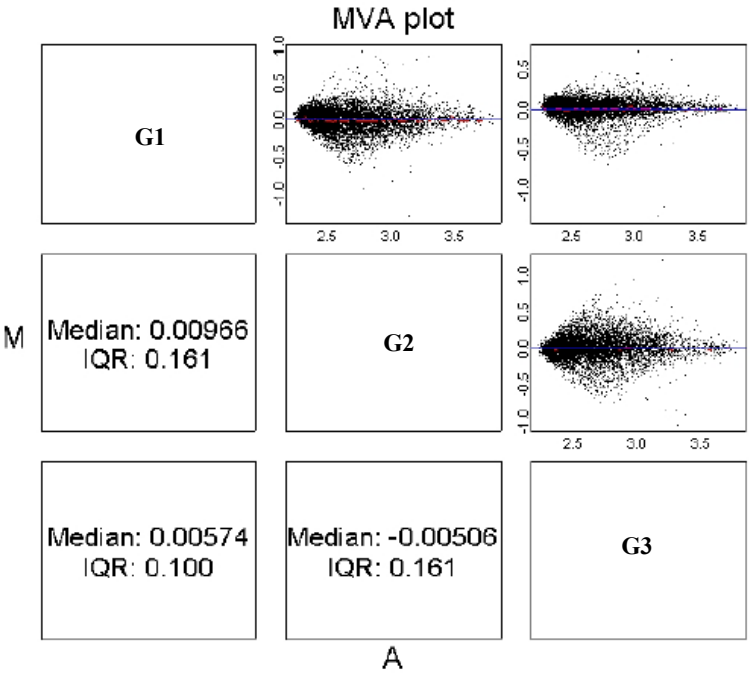
```
R>write.exprs(eset, file="mydata.txt")
```

6. Построение диагностических графиков, показывающих эффективность проведенной стандартизации данных

Аналогично пункту 4.

```
R>boxplot(eset,col=c(2,2,2,2,2,2,3,3,3,3,3,3))
```





7. Аннотация

Для облегчения восприятия результата создадим таблицу с описанием, включающую имена соответствующих генов, описание и, возможно, другую информацию из баз данных.

Включение столбца Probe в таблицу

```
R> col<-aaf.handler()[[1]]
```

Создание таблицы:

```
R> table<-aafTableAnn(affyID[1:12625],"hgu95av2”,col)
```

Описание генов:

```
R> descr<-multiget(affyID[1:12625],
env=hgu95av2GENENAME)
```

Создаем второй столбец с описанием генов:

```
R> col2<-aafTable(«Description»=descr)
```

Сливаем этот столбец с предыдущей таблицей:

```
R> table2<-merge(table, col2)
```

8. Определение генов, изменивших уровень своей экспрессии при сравнении двух различных экспериментов

Маркируем принадлежность арреев к одной из 2 групп:

```
R>data.c<-c(0,0,0,0,0,0,1,1,1,1,1,1)
```

Вычисляем t-статистику:

```
R>teststat<-mt.teststat(eset,data.c)
```

Вычисляем нескорректированное значение p:

```
R> rawp0<-2*(1-pnorm(abs(teststat)))
```

Коррекция значения P с использованием методов Бонферони, Холма, Шидака и т. д.

```
R>procs<-c("Bonferroni","Holm","Hochberg","SidakSS",
"SidakSD", "BH","BY")# методы коррекции
R>res<-mt.rawp2adjp(rawp0,procs)
R>adjp<-res$adjp[order(res$index),]#матрица с
количеством генов, изменивших экспрессию для
скорректированных значений p
R> mt.reject(adjp,seq(0,1,0.1))$r   # выводим матрицу
на экран
```

	rawp	Bonferroni	Holm	Hochberg	SidakSS	SidakSD	BH	BY
0	1	1	1	1	1	1	1	1
0.1	2771	97	97	97	98	98	766	220
0.2	4068	116	118	118	123	123	1291	305
0.3	5232	135	135	135	140	141	1781	359
0.4	6300	143	143	143	148	149	2416	425
0.5	7430	148	148	148	160	161	3107	474
0.6	8463	156	156	156	175	176	3903	546
0.7	9514	161	163	163	184	186	4908	602
0.8	10593	167	168	168	197	198	6265	658
0.9	11632	174	175	175	222	222	8553	713
1	12625	12625	12625	12625	12625	12625	12625	12625

Первый столбец — уровень значимости, второй — нескорректированные значения *p*, все остальные — значения *p*, скорректированные различными методами.

```
R>which<-mt.reject(adjp,0.01)$which[,2]# отбор с уров-
нем значимости 0.01 и по методу Bonferroni
R>  results<-table2[which,2]  #  отбираем  гены  по  2
столбцу таблицы с аннотацией
R> saveText(results,"sorted_bonf.txt")
```

Альтернативный вариант коррекции — коррекция методом перестановок:

```
R>resT<-mt.maxT(eset,data.c,B=1000)
R>ord<-order(resT$index)
R>rawp<-resT$rawp[ord]
R>maxT<-resT$adjp[ord]
R>teststat<-resT$teststat[ord]
R>mt.reject(cbind(rawp,maxT),seq(0,1,0.1))$r
```

	rawp	maxT
0	0	0
0.1	2462	6
0.2	3731	14
0.3	4909	19
0.4	6009	27
0.5	7111	35
0.6	8224	40
0.7	9309	52
0.8	10430	70
0.9	11563	91
1	12625	12625

```
R>test.maxT<-
mt.reject(cbind(rawp,maxT),seq(0,1,0.1))$r
R>write.table(test.maxT,file="mas_maxT.txt")
# записываем результаты в таблицу
R> which<-mt.reject(cbind(rawp,maxT),0.1)$which[,2]
R>results1<-table2[which,2]
R> saveText(results,"sorted_maxT.txt")
```

В результате с помощью метода Бонферони мы получили список из 57 генов, с достоверностью 99% изменивших свою экспрессию.

a disintegrin and metalloproteinase domain 22
A kinase (PRKA) anchor protein 1
adaptor-related protein complex 3, beta 2 subunit
adenylate kinase 2
ankyrin 3, node of Ranvier (ankyrin G)
apoptosis inhibitor 5
ATPase, aminophospholipid transporter (APLT), Class I, type 8A, member 1
ATP-binding cassette, sub-family C (CFTR/MRP), member 8
B/K protein
bassoon (presynaptic cytomatrix protein)
brain-specific protein p25 alpha
breakpoint cluster region
casein kinase 1, delta
centromere protein E, 312kDa
choline kinase-like
chromogranin A (parathyroid secretory protein 1)
chromogranin B (secretogranin 1)
creatine kinase, mitochondrial 1 (ubiquitous)
down-regulator of transcription 1, TBP-binding (negative cofactor 2)
Fas apoptotic inhibitory molecule 2
FK506 binding protein 12-rapamycin associated protein 1
G protein-coupled receptor 19
guanine nucleotide binding protein (G protein), gamma 5
H326
hypothetical protein DKFZp761N09121
hypothetical protein LOC157627
integrin beta 3 binding protein (beta3-endonexin)
internexin neuronal intermediate filament protein, alpha
KDEL (Lys-Asp-Glu-Leu) endoplasmic reticulum protein retention receptor 1
Kruppel-like factor 4 (gut)
LIM domain only 6
microtubule-associated protein 1A
MUF1 protein

p21 (CDKN1A)-activated kinase 3
peptidase D
phosphodiesterase 8B
potassium voltage-gated channel, KQT-like subfamily, member 2
protein kinase C, beta 1
protein phosphatase 1, regulatory (inhibitor) subunit 12B
protein tyrosine phosphatase, receptor type, N
protein tyrosine phosphatase, receptor type, N
protein tyrosine phosphatase, receptor type, N polypeptide 2
pyruvate dehydrogenase complex, component X
RaP2 interacting protein 8
ras homolog gene family, member C
regulating synaptic membrane exocytosis 2
ribosomal protein S12
SH3-domain GRB2-like 3
solute carrier family 31 (copper transporters), member 1
sorting nexin 7
synaptosomal-associated protein, 25kDa
synaptosomal-associated protein, 91kDa homolog (mouse)
transcription elongation factor A (SII)-like 1
transient receptor potential cation channel, subfamily C, member 1
ubiquitin-conjugating enzyme variant Kua
uroporphyrinogen III synthase (congenital erythropoietic porphyria)
uroporphyrinogen III synthase (congenital erythropoietic porphyria)

ЗАКЛЮЧЕНИЕ

В данной статье был рассмотрен подход к решению задачи выделения генов, изменивших с определенной достоверностью уровень своей экспрессии на основе данных из нескольких микрочиповых экспериментов.

Описан процесс обработки данных с олигонуклеотидного чипа компании Affimetrix. В эксперименте изучалась экспрессия генов в клетках раковых опухолей мозга глиобластомы и олигобластомы.

Нормализация и кластеризация данных выполнялись в среде программирования языка *R GUI* с использованием пакетов *Bioconductor*. В резуль-

тате был получен список из 57 генов, изменивших свою экспрессию с достоверностью 99%.

СПИСОК ЛИТЕРАТУРЫ

1. **Irizarry R.A., Hobbs B., Collin F., et al.** Exploration, normalization, and summaries of high density oligonucleotide array probe level data // Biostatistics. — 2003. — Vol. 4, No. 2. — P. 249–264.
2. **Bolstad B.M., Irizarry R.A., Astrand M., Speed T.P.** A comparison of normalization methods for high density oligonucleotide array data based on variance and bias // Bioinformatics. — 2003. — Vol. 2, N. 19. — P.185–193.