

bioconductor

Дифференциальная экспрессия
генов по данным RNA-seq.
Геномные интервалы

Артем Артемов

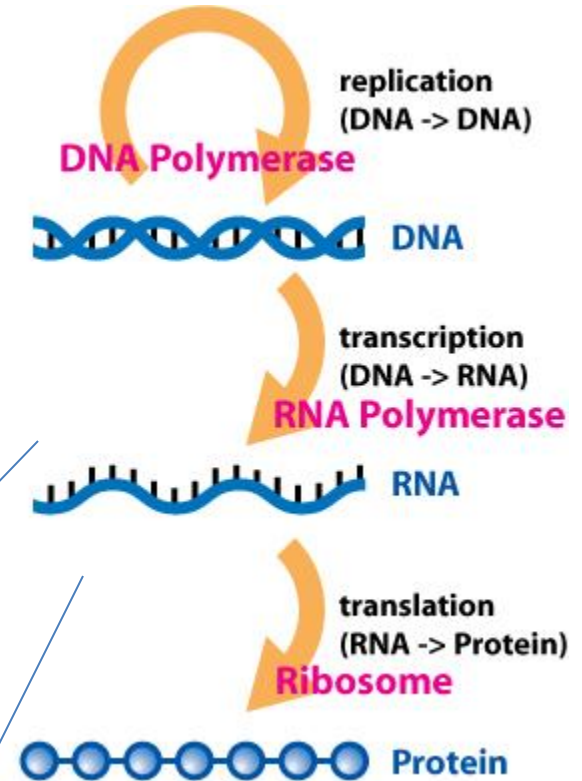
23.04.2014

Outline

- Биологическая задача
 - ДНК – РНК – белок
 - Next Generation Sequencing
 - RNA-seq
- Набор пакетов bioconductor
- Статистическая модель
 - Редкие события и распределение Пуассона
 - Проблема овердисперсии и отрицательное биномиальное распределение
 - Пакет DESeq (и edgeR)
- Работа с геномными интервалами в bioconductor

Немного биологии

- «Центральная догма молекулярной биологии»
- Оцениваем уровень экспрессии каждого гена по количеству соответствующей РНК
- Существенное дополнение для эукариот – сплайсинг. Интроны вырезаются, **ЭКЗОНЫ** сшиваются между собой



Pre-mRNA

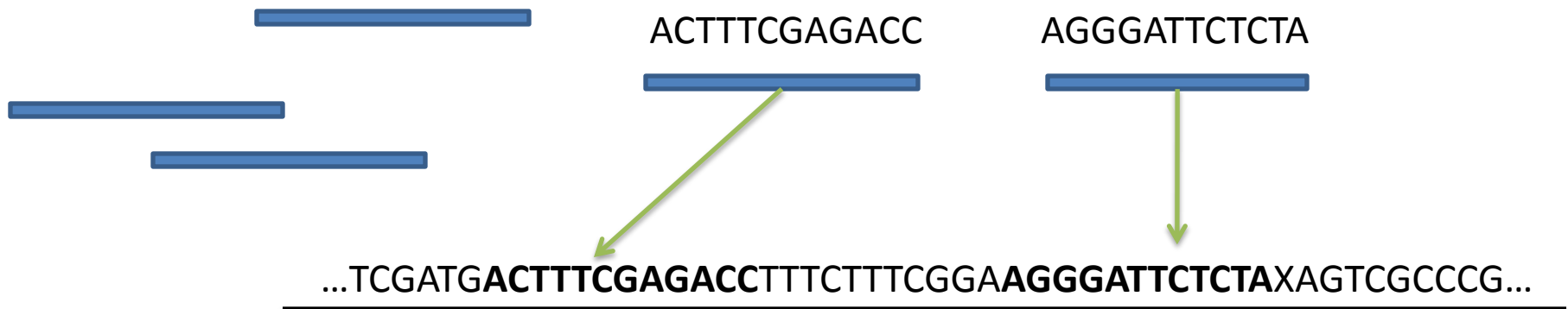


Mature mRNA

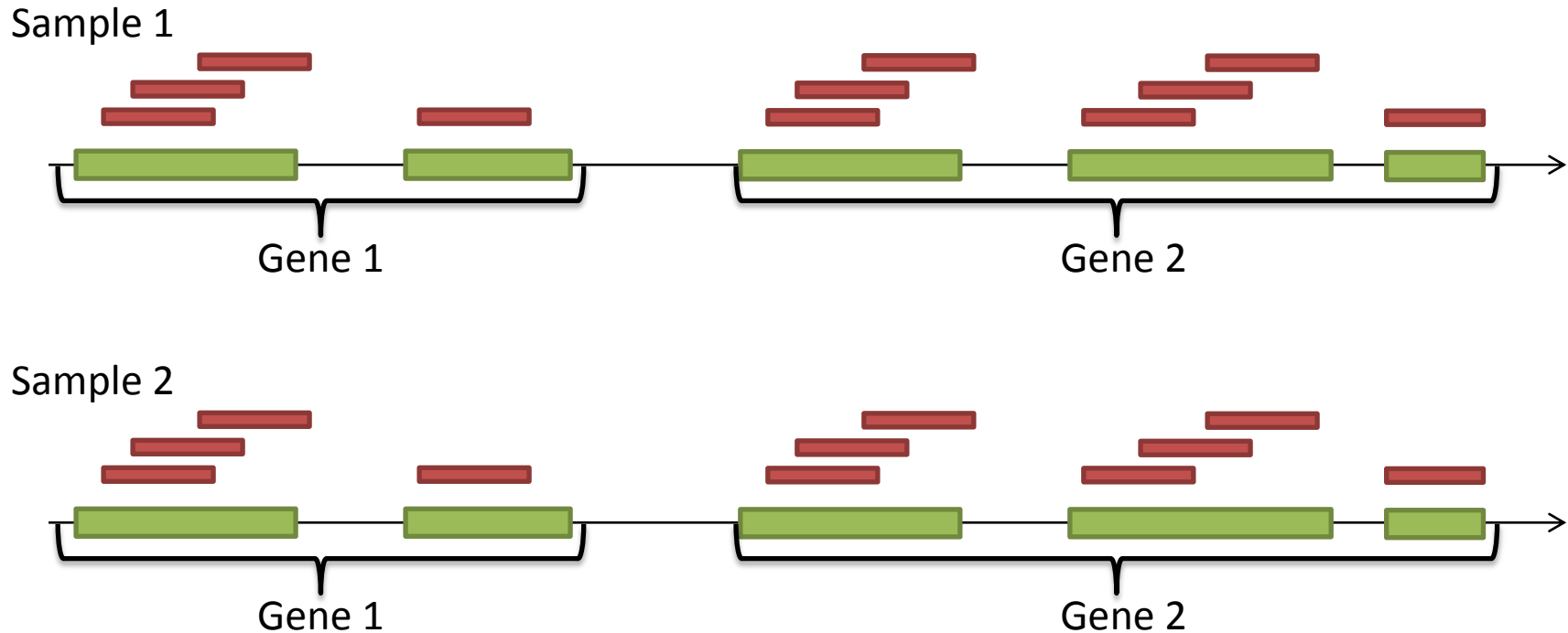


Секвенирование и маппирование

- Упрощенно: берем всю РНК, режем на кусочки, прочитываем (=секвенируем) какую-то часть этих кусочков
- Выравнивание (маппирование) = Alignment (mapping). Смотрим на последовательность каждого кусочка (рид=read, 75-100 букв) и на последовательность генома, находим в геноме (почти) идентичную последовательность



Экспрессия генов



- Вспоминаем про разметку генома на гены. Будем рассматривать случай, когда знаем координаты генов и экзонов в них.

Входные данные

- Таблица с количеством ридов на каждый ген в каждом образце.

> `head(countTable)`

Гены	Образцы			
	Контроли		После воздействия	
	untreated3	untreated4	treated2	treated3
FBgn00000003	0	0	0	1
FBgn00000008	76	70	88	70
FBgn00000014	0	0	0	0
FBgn00000015	1	2	0	0
FBgn00000017	3564	3150	3072	3334
FBgn00000018	245	310	299	308

...

Про установку пакетов

- Обычно пакеты устанавливаются из центрального репозитория:

```
install.packages("название")
```

- bioconductorR – самостоятельный репозиторий

```
source("http://www.bioconductor.org/biocLite.R")
```

то же, что и загрузка скрипта, только из интернета

```
biocLite("DESeq")      # загрузка пакета
```

```
library(DESeq)
```

Поправка на количество прочитанных ридов

- Очевидная причина отличий – разное суммарное количество ридов в каждом образце
- Самый простой выход – поделить количество ридов для каждого гена на общее количество ридов в образце
- RPM: reads per million mapped reads
- RPKM: reads per kilobase per million mapped reads

$$RPM = \frac{10^6 k_{ij}}{N_j}; \quad RPKM = \frac{10^9 k_{ij}}{N_j L}$$

k_{ij} – количество ридов в образце j для гена i , N_j – общее кол-во ридов в образце j , L – длина гена

Поправка на количество прочитанных ридов

- Хотим корректировать (делить количество ридов на ген на поправочный коэффициент для данного образца s_j) так, чтобы новые значения были тех же порядков, что и старые
- Например, так:
 $s_j = (\text{среднее по образцу}) / (\text{среднее по всей таблице})$
- Проблема: изменение экспрессии высокоэкспрессирующихся генов слишком сильно влияет на общую сумму.
- Выход: оценим поправку каждого образца для каждого гена по отдельности (пусть неточно), затем найдём медианную поправку.

$$r_{ij} = k_{ij} / \text{CP_ГЕОМ}_{\text{по_рядам}}(k_{ij}); \quad s_j = \text{median}(r_{ij})$$

Поправка на количество прочитанных ридов

Пример 1

	s1	s2
ген1	10	20
ген2	15	30
ген3	100	200
	: 2/3	: 4/3

Пример 2

	s1	s2
ген1	10	10
ген2	10	10
ген3	100	220
	: 2/3	: 4/3

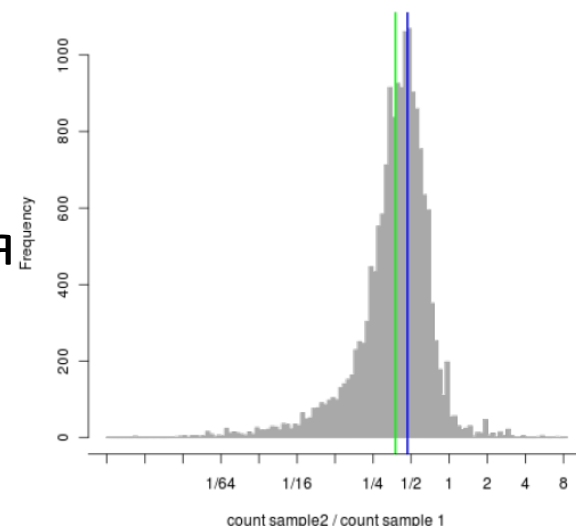
эти гены будут дифф. экспр (хотя не меняют экспрессию)

эти гены меняются экспрессию, но не будут детектированы

$sj = (\text{среднее по образцу}) / (\text{среднее по всей таблице})$

Выход: оценим поправку каждого образца для каждого гена по отдельности (пусть неточно), затем найдём медианную поправку.

$rij = kij / \text{CP_ГЕОМпо_рядам}(kij); \quad sj = \text{median}(rij)$



Поправка на количество прочитанных ридов

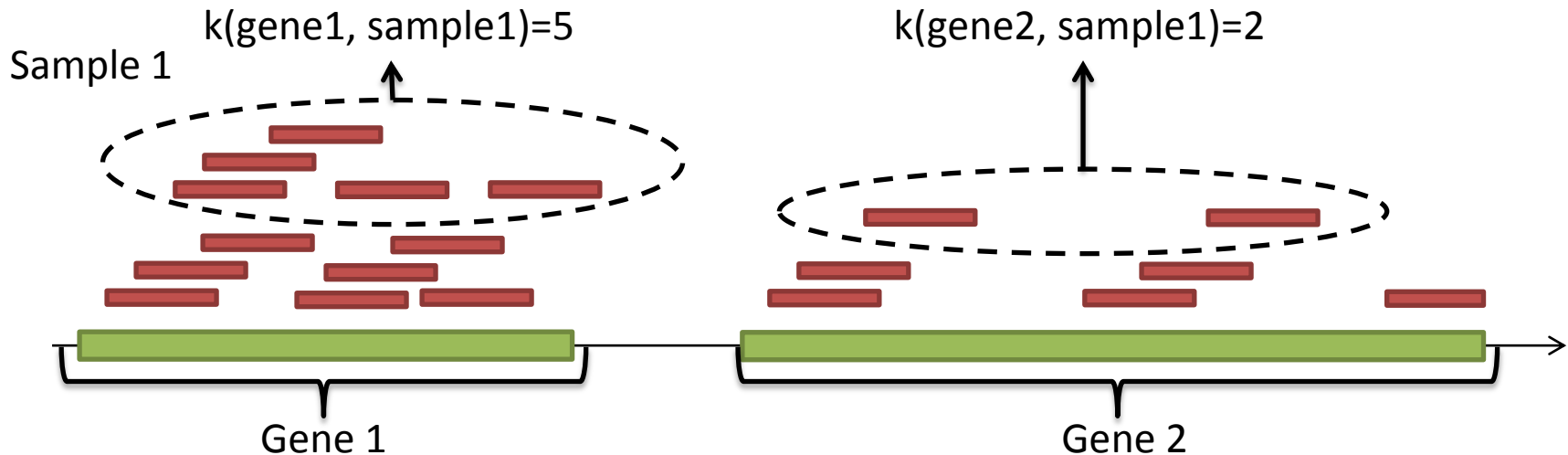
```
> condition = factor( c( "untreated", "untreated",  
  "treated", "treated" ) )  
> cds = newCountDataSet( countTable, condition )
```

```
> cds = estimateSizeFactors( cds )  
> sizeFactors( cds )  
untreated3 untreated4 treated2 treated3  
0.8730966 1.0106112 1.0224517 1.1145888
```

```
> countsNorm= counts( cds, normalized=TRUE )  
               untreated3 untreated4 treated2 treated3  
FBgn0000003      0.000000  0.000000  0.000000  0.8971919  
FBgn0000008    87.046493  69.26502  86.06763  62.8034302  
FBgn0000014      0.000000  0.000000  0.000000  0.0000000  
FBgn0000015      1.145349  1.97900  0.000000  0.0000000  
FBgn0000017  4082.022370 3116.92579 3004.54278 2991.2376629  
FBgn0000018   280.610404  306.74508  292.43434  276.3350930
```

Модель

- Секвенируем, поймем, куда в геноме попадает каждый рид, посчитаем, сколько ридов попадает в каждый ген
- Каждый ген \rightarrow много фрагментов РНК. Мы прочитываем только часть из них



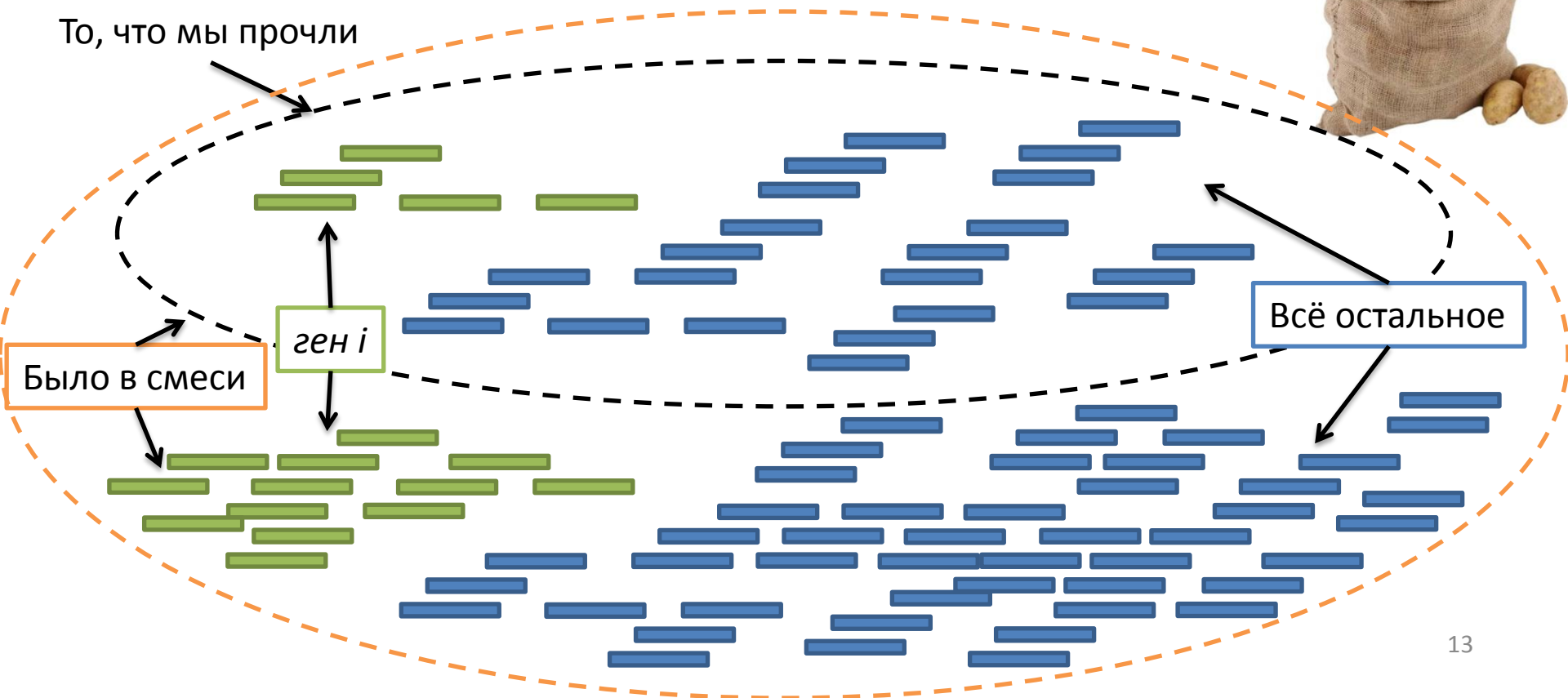
- Предполагаем, что количество ридов $k(\text{gene } i, \text{sample } i)$ пропорционально реальному количеству фрагментов РНК данного гена в данном образце.

Модель

- Посмотрим на один *ген i*
- На него упало k ридов, остальные риды ($N-k$) упали на другие гены, или вообще в межгенные области
- Аналогия: мешок с **зелеными** и **синими** шарами. Вытащили из него N случайных шаров, какая вероятность, что из них k зеленых, если доля зеленых шаров в мешке p



То, что мы прочли



Биномиальное распределение?

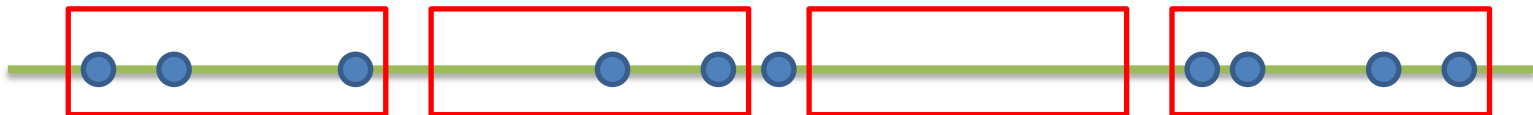
- Прочли фрагментов РНК гораздо меньше, чем было в смеси (= вытащили шаров меньше, чем было в мешке).
- Вероятность того, что среди N вытащенных шаров зелеными окажутся k (если вероятность вытащить зеленый шар p)

$$P(k) = C_N^k p^k (1-p)^{(N-k)}$$

- Но, $N \gg k$ (много больше), например:
 k в интервале от 10 до 100 тысяч
 N в интервале от 9 миллионов до 100 миллионов

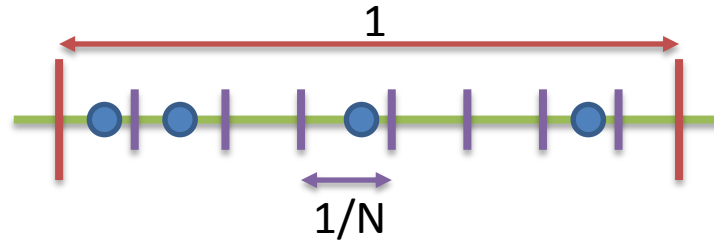
Распределение Пуассона

- Распределение количества редких событий в единицу времени (расстояния, объема) при ожидаемой интенсивности λ
 - сколько автобусов проехало мимо за единицу времени, если вы ожидаете увидеть λ автобусов
 - сколько человек проголосовало за единицу времени
 - сколько изюминок в булочке в единице объема



В среднем, в интервал попадает 3 точки, но могут быть и 2, и 0, и 4

Распределение Пуассона – вывод



- Предел Биномиального распределения
- Разобьем наш интервал (длины 1) на N одинаковых интервалов (длины $1/N$), настолько маленькие, что события в них происходят настолько редко, что либо не происходят, либо происходят единожды
- Вероятность того, что событие произойдет в маленьком интервале $p = \lambda/N$
- Какова вероятность, что событие произойдет k раз в большом интервале

$$P(k) = C_N^k p^k (1-p)^{(N-k)} = C_N^k \left(\frac{\lambda}{N}\right)^k \left(1 - \frac{\lambda}{N}\right)^{(N-k)} \xrightarrow{N \rightarrow \infty} \frac{\lambda^k}{k! e^{-\lambda}}$$

$$C_N^k = \frac{n!}{k!(n-k)!}; \quad n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \quad \leftarrow \text{(Формула Стирлинга)}$$

Распределение Пуассона

- При таких соотношениях k и N **Пуассон** – очень хорошая аппроксимация **биномиального** распр.
- Эксперимент: построим функцию вероятностей для этих двух распределений

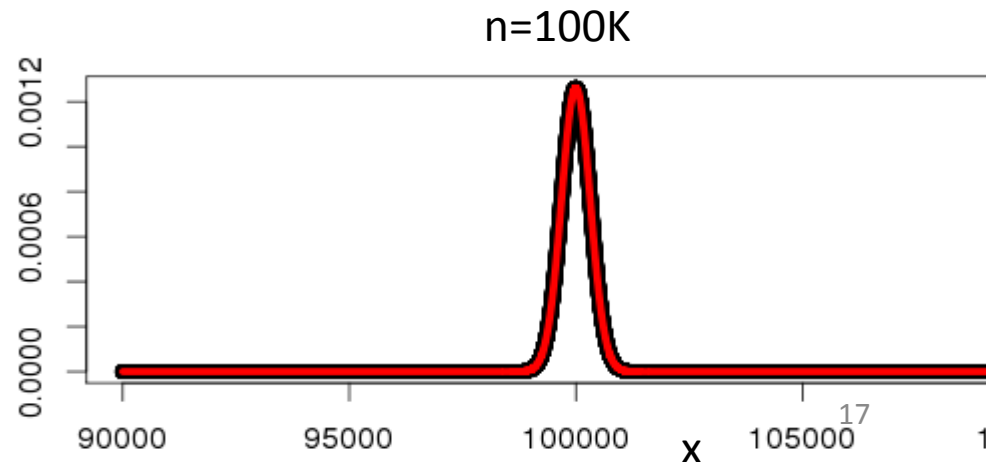
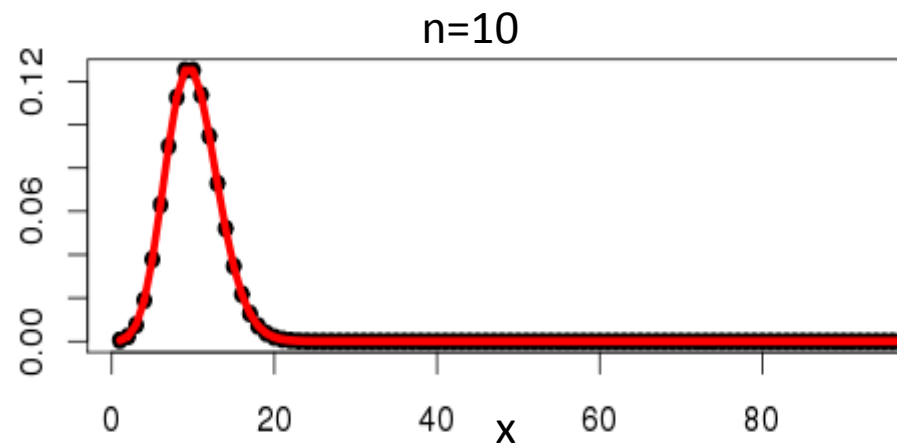
$x=1:100$ ИЛИ $x=90000:110000$

$N=9e9$ #всего прочли 9 миллионов ридов

$n=10$ ИЛИ $n=1e5$ #10 или 100K ридов на ген

`plot(x, dbinom(x, size=N, prob=n/N), pch=19)`

`lines(x, dpois(x, n), lwd=5, col="red")`



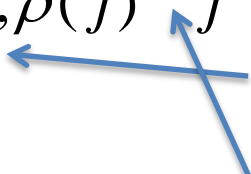
Дифференциальная экспрессия

- Подумаем, как бы мы могли искать дифференциально экспрессирующиеся гены, основываясь на распределении Пуассона
- Для каждого гена i построим модель

$$k_{ij} \sim \text{Pois}(\mu_{ij}) \Rightarrow E(k_{ij}) = \mu_{ij}; \quad D(k_{ij}) = \mu_{ij}$$

$$\mu_{ij} = \mu_{i,\rho(j)} s_j$$

Средняя экспрессия данного гена =
средняя экспрессия для данного состояния
(ρ = больной, контроль и т.д.) * поправочный
коэффициент (разное количество ридов в образцах)



Дифференциальная экспрессия

- При нулевой гипотезе: $\mu_{i,\rho_1} = \mu_{i,\rho_2}$
- Итого, наши действия: оценим среднюю экспрессию каждого гена. Она же (если верить в распределение Пуассона) – дисперсия
- Можем проверить, отличаются ли μ_{i,ρ_1} и μ_{i,ρ_2}

Овердисперсия

- Технические реплики – один и тот же биологический образец обработали и секвенировали два раза
- Биологические реплики – взяли два разных образца, обработали и секвенировали
- Дисперсии в распределении Пуассона достаточно, чтобы объяснить отличия между техническими репликами
- Биологические реплики отличаются сильнее (отличаются не только сколько мы ридов на каждый ген прочли, но и сколько таких фрагментов РНК изначально было) – овердисперсия
- Но в распределении Пуассона **Дисперсия=Среднее**
- Выход: взять другое распределение имеющее 2 параметра, такое, что Пуассон – его частный случай

Отрицательное биномиальное распределение

- Отрицательное биномиальное (negative binomial) распределение
- зафиксируем количество неудач r . Как распределено кол-во успехов Y
- $Y \sim \text{NB}(r, p)$ $\mathbb{P}(Y = k) = \binom{k+r-1}{k} p^r q^k, \quad k = 0, 1, 2, \dots$
- Два параметра вместо одного, можем их подобрать так, чтобы распределение имело нужные среднее μ и дисперсию $\mu + \delta$
- Распределение Пуассона – частный случай

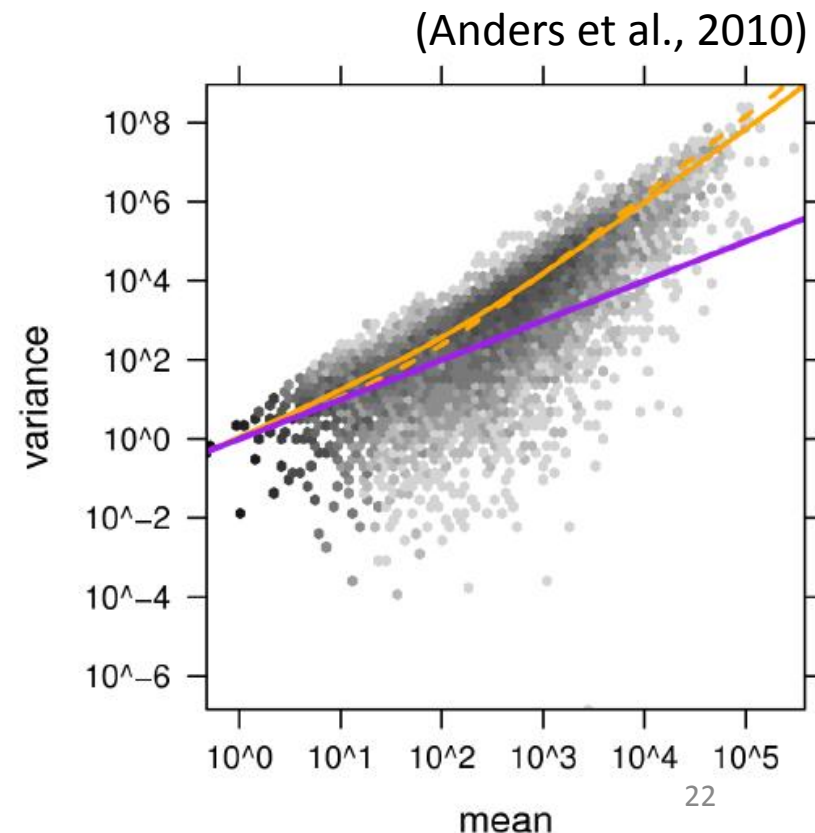
$$\text{Poisson}(\lambda) = \lim_{r \rightarrow \infty} \text{NB}\left(r, \frac{\lambda}{\lambda + r}\right).$$

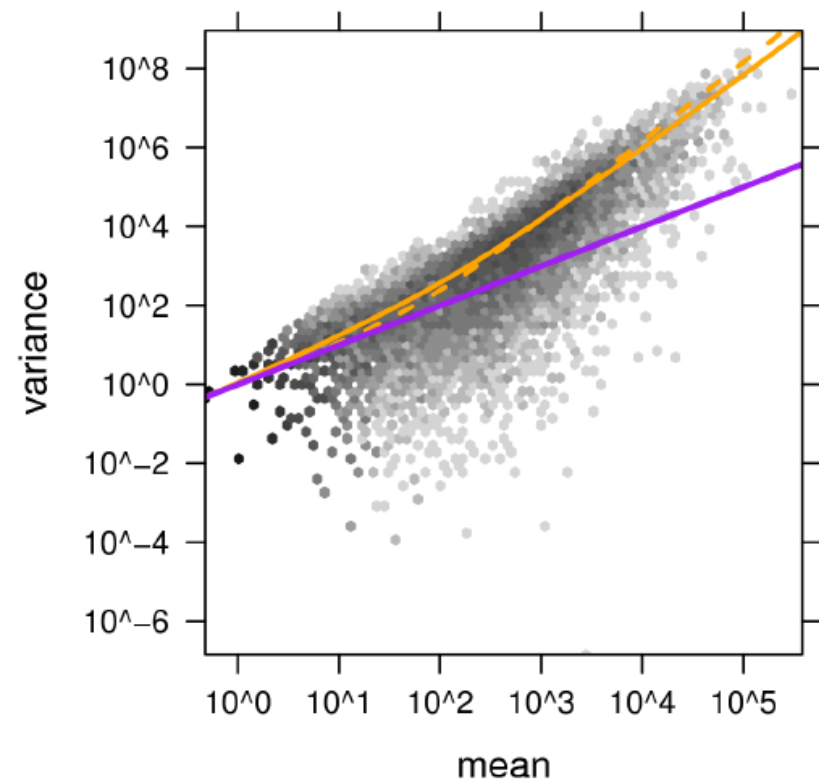
Как оценить дисперсию для каждого гена?

- Проблема: раньше дисперсию для каждого гена легко было оценить по среднему значению. Как теперь?

Фиолетовая кривая – Пуассоновская модель: $var=mean$

- В идеале: много биологических реплик, для каждого гена и каждого состояния – много чисел, оценим дисперсию
- Обычно реплик мало (2-3). Предположение: дисперсия всё равно как-то зависит от среднего (но не обязательно линейно)
- DESeq: построим по всем генам **локальную регрессию дисперсии от среднего**





Отрицательное биномиальное
(negative binomial) распределение
зафиксируем количество неудач r .
Как распределено кол-во успехов
NB (mean = μ_i , var = $\mu_i + \delta_i$)
 $k_{ij} \sim \text{NB} \text{ (mean = } \mu_i, \text{ var = } \mu_i + \delta_i \text{)}$

Variance calculated from comparing two replicates

Poisson

$$v = \mu$$



(Anders et al., 2010)

Poisson + constant CV

$$v = \mu + \alpha \mu^2$$



Poisson + local regression

$$v = \mu + f(\mu^2)$$



EMBL



DESeq

#оценим дисперсию

```
cds = estimateDispersions( cds )
```

#собственно, тест

```
res = nbinomTest( cds, "untreated", "treated" )
```


Результат

Упорядочим по
возр. p-value

```
> head(res[order(res$padj), ])
```

	id	baseMean	baseMeanA	baseMeanB	foldChange	log2FoldChange
	FBgn0039155	463.4369	884.9640	41.90977	0.0473576	-4.400260
	FBgn0025111	1340.2282	311.1697	2369.28680	7.6141316	2.928680
	FBgn0003360	2544.2512	4513.9457	574.55683	0.1272848	-2.973868
	FBgn0029167	2551.3113	4210.9571	891.66551	0.2117489	-2.239574
	FBgn0039827	188.5927	357.3299	19.85557	0.0555665	-4.169641
	FBgn0035085	447.2485	761.1898	133.30718	0.1751300	-2.513502

	pval	padj
	1.641210e-124	1.887556e-120
	3.496915e-107	2.010901e-103
	1.552884e-99	5.953239e-96
	4.346335e-78	1.249680e-74
	1.189136e-65	2.735251e-62
	3.145997e-56	6.030352e-53

Средние
по группам

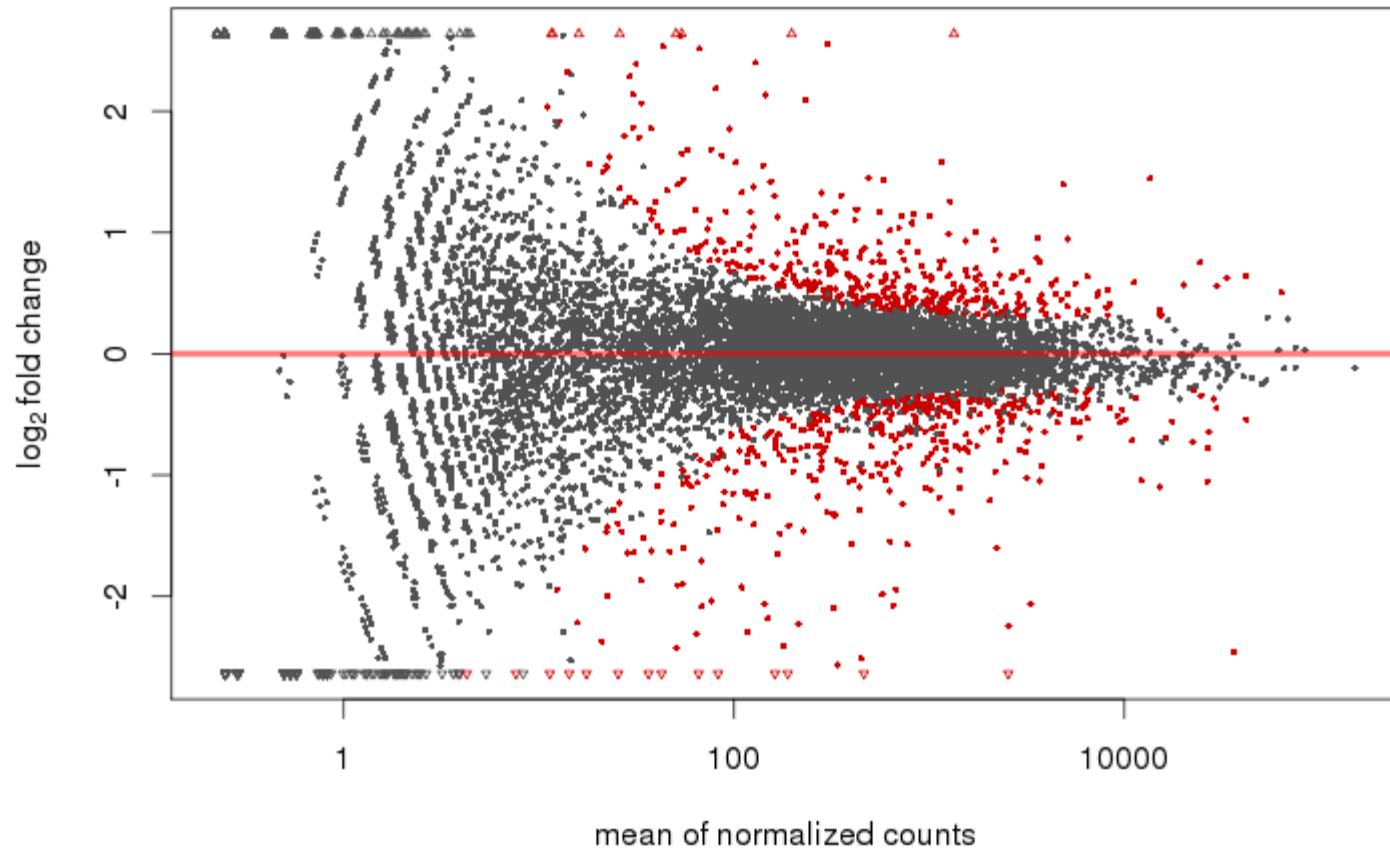
Fold change:
отношение средних
(и его логарифм)

P-value

P-value, скорректированное на
множественное тестирование

Нарисуем

```
> plotMA(res)
```



Парный тест и несколько факторов

counts						
	s1	s2	s3	s4	s5	s6
gene1						
gene2						
gene3						
...						

design		condition	patient
	s1	normal	1
	s2	tumor	1
	s3	normal	2
	s4	tumor	2
	s5	normal	3
	s6	tumor	3

- Линейная модель (для каждого гена)
- Содержательный случай – парные образцы (например, образцы здоровой и пораженной ткани из одного человека)

```
> cdsFull = newCountDataSet( counts, design )  
> cdsFull = estimateSizeFactors( cdsFull )  
> cdsFull = estimateDispersions(cdsFull)  
> fit1 = fitNbinomGLMs( cdsFull, count ~ condition + patient )  
> fit0 = fitNbinomGLMs( cdsFull, count ~ patient )  
> pvals=nbinomGLMTest ( fit1, fit0 ) #получаем вектор p-values  
> pvals.adj=p.adjust(pvals, method="BH")
```

Работа с отрезками в R

- Пожалуй, самое частое, что приходится делать в геномике. Ген, экзон, сайт связывания транскрипционного фактора, CpG-остров – суть отрезки в геноме, имеют координаты (хромосома – начало – конец)
- В общем случае, задача пересечения двух систем отрезков алгоритмически нетривиальна (если отрезки в одной из систем могут пересекаться). Структура данных «интервальное дерево»
- Задача: найдём такие участки генома, куда попадают в достаточном количестве риды РНК, такие, что эти участки не пересекаются с генами.

Хранилище для отрезков

- IRanges
 - координаты начала, конца и любые другие поля
- GenomicRanges
 - Хранит дополнительно название хромосомы (seqnames) и направление, или «цепь» (strand, + / -)

Загрузка разметки генов

- Загрузим таблицу с координатами экзонов всех известных генов Дрозофилы

```
biocLite("biomaRt")
```

```
library(biomaRt)      #http://www.ensembl.org/biomart/martview
```

```
#какую базу данных использовать
```

```
ensembl <- useMart("metazoa_mart_16",  
  dataset="dmelanogaster_eg_gene")
```

```
#какие поля вытащить
```

```
fields <- c("chromosome_name","strand","ensembl_gene_id",  
  "ensembl_exon_id","start_position",  
  "end_position","exon_chrom_start","exon_chrom_end")
```

```
genes <- getBM( fields, mart=ensembl, filters="chromosome_name",  
  values=c("4"))
```

Загрузка разметки генов

```
> head( genes )
```

	chromosome_name	strand	ensembl_gene_id	ensembl_exon_id
1	4	1	FBgn0040037	FBgn0040037:1
2	4	1	FBgn0040037	FBgn0040037:2
3	4	1	FBgn0040037	FBgn0040037:3
4	4	1	FBgn0040037	FBgn0040037:4
5	4	-1	FBgn0052011	FBgn0052011:6
6	4	-1	FBgn0052011	FBgn0052011:5

	start_position	end_position	exon_chrom_start	exon_chrom_end
1	24053	25665	24053	24477
2	24053	25665	24979	25153
3	24053	25665	25218	25450
4	24053	25665	25501	25665
5	26455	32391	29356	32391
6	26455	32391	28966	29301

Создадим из этой таблицы набор отрезков (GenomicRanges)

Создадим GenomicRanges

```
annot <- GRanges(  
  seqnames = Rle("chr4"), #хромосома  
  ranges=IRanges(  
    start=genes$exon_chrom_start,  
    end=genes$exon_chrom_end  
  ), #объект IRanges  
  strand = Rle(genes$strand),  
  exon=genes$ensembl_exon_id,  
  gene=genes$ensembl_gene_id  
  #все остальные поля  
)
```


Загрузка bam-файла с выравниваниями ридов на геном

```
> biocLite("Rsamtools")
> library("Rsamtools")
> aln_all <- readGappedAlignments("...../untreated1_chr4.bam")
> head(aln_all)
```

GappedAlignments with 6 alignments and 0 metadata columns:

	seqnames	strand	cigar	qwidth	start
<Rle>	<Rle>	<character>	<integer>	<integer>	
[1]	chr4	-	75M	75	892
[2]	chr4	-	75M	75	919
[3]	chr4	+	75M	75	924

	end	width	ngap
	<integer>	<integer>	<integer>
[1]	966	75	0
[2]	993	75	0
[3]	998	75	0

seqlengths:

chr2L	chr2R	chr3L	...	chrM	chrX	chrYHet
23011544	21146708	24543557	...	19517	22422827	34703

Посчитаем покрытие – способ 1

- Посчитаем для каждого отрезка – экзона, сколько отрезков – ридов с ним пересекаются

```
> overlaps=countOverlaps( annot, aln_all )
```

```
> head(overlaps)
```

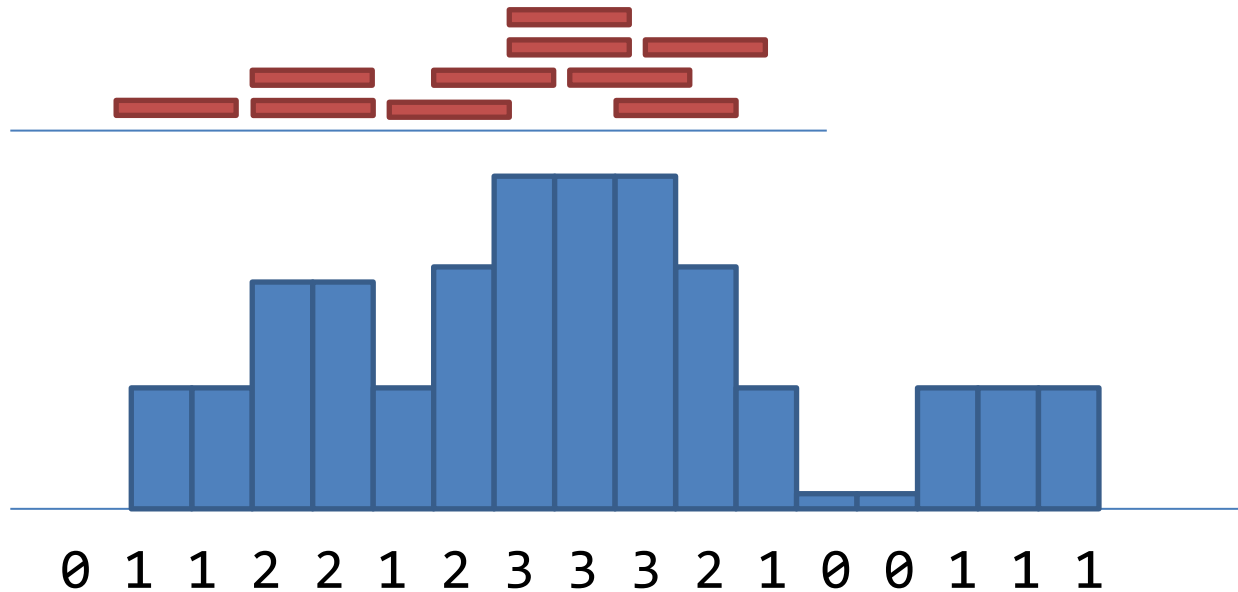
```
[1]      0      0      0      0 774    98
```

- Далее сложим получившиеся значения для всех экзонов каждого гена. `tapply`

```
> tapply(overlaps, list( annot$gene ), FUN=sum )
```

```
FBgn0002521  FBgn0004607  FBgn0004859  FBgn0010217  
          410              7          373          12513
```

Rle и покрытие



- Run length encoding:

2 раза по 1, 2 раза по 2, 1 раз по 1, 1 раз по 2, 3 раза по 3, ...

```
> Rle(c(1,1,2,2,1,2,3,3,3,2,1,0,0,1,1,1))
```

```
numeric-Rle of length 16 with 9 runs
```

```
Lengths: 2 2 1 1 3 1 1 2 3
```

```
Values : 1 2 1 2 3 2 1 0 1
```

Покрытие

```
> coverage(aln_all)
```

```
SimpleRleList of length 8
```

```
$chr2L
```

```
integer-Rle of length 23011544 with 1 run
```

```
Lengths: 23011544
```

```
Values :    0
```

```
.....
```

```
$chr4
```

```
integer-Rle of length 1351857 with 122061 runs
```

```
Lengths: 891  27  5  12  13  45  5 ...  1  3 106  75 1600  75 1659
```

```
Values :  0  1  2  3  4  5  4 ... 10  6  0  1  0  1  0
```

```
> cvr=coverage(aln_all)
```

```
> cvr_chr4=cvr$chr4
```

Список, каждый элемент соответствует
хромосоме и является Rle

aggregate для отрезков

Беда в том, что работает только с IRanges (не GRanges). Придётся пройтись циклом по хромосомам

```
> cvr_chr4=cvr$chr4
```

```
> annot_chr4_IR=annot[seqnames(annot)=='chr4', ]@ranges
```

annot_chr4_IR – уже IRanges

```
> annot_chr4_IR
```

IRanges of length 1069

	start	end	width
[1]	24053	24477	425
[2]	24979	25153	175
[3]	25218	25450	233
[4]	25501	25665	165

Дополнительный вопрос:
как таким способом
посчитать количество
ридов на экзон?

Получили среднее
покрытие экзонов

```
> aggregate(cvr_chr4, annot_chr4, FUN=mean)
```

```
[1] 1.764706e-01 0.000000e+00 3.763076e+01
```

Заметьте, что среднее покрытие и количество упавших на экзон ридов – разные вещи:



3 рида,
покрытие $\sim 3 \times \text{длина рида} / \text{длина экзона}$

Внимание: S4-объекты

- С большинством объектом в R можно работать как со списками:

`x$p.value`

`names(x)`

- bioconductorR использует S4 объекты.
Аналогичные конструкции:

`x@field`

`slotNames(x)`

Что ещё почитать – RNA-seq

Курс лекций про анализ данных NGS на ФББ МГУ (+видео)

<http://bioinf.fbb.msu.ru/wiki/index.php/NgsCourse>

DESeq

- <http://www.bioconductor.org/help/course-materials/2011/RNASeqChIPSeq/Lectures/RNASeq-DifferentialExpression-SimonAnders.pdf>
- “Dealing with aligned data: mapping, expression estimation, normalisation, DE” Mar González-Porta
- <http://www.bioconductor.org/packages/2.12/bioc/vignettes/DESeq/inst/doc/DESeq.pdf>
- Anders and Huber, 2010. Differential expression analysis for sequence count data
<http://genomebiology.com/content/11/10/R106>

edgeR

- <http://bioconductor.org/packages/2.12/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf>
- Robinson et al., 2010
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2796818/>

Comparisons

- <http://davetang.org/muse/2011/01/05/deseq-vs-edger-vs-bayseq/>

Что ещё почитать – GenomicRanges

- <http://www.biostat.jhsph.edu/~khansen/IRangesLecture.pdf>
- <http://bioconductor.org/>

Shot noise и распределение Пуассона

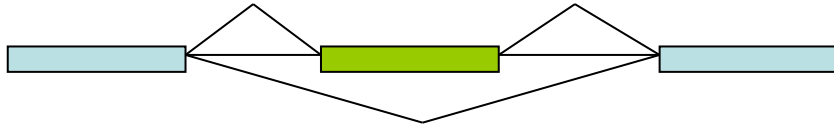
- Пусть в одном образце увидели 5 фрагментов на данный ген, насколько удивительно в другом образце увидеть 6 фрагментов

С какими данными приходится работать

- bam
- bed
- bedGraph, wig, bigWig

Differential splicing

Pre-mRNA



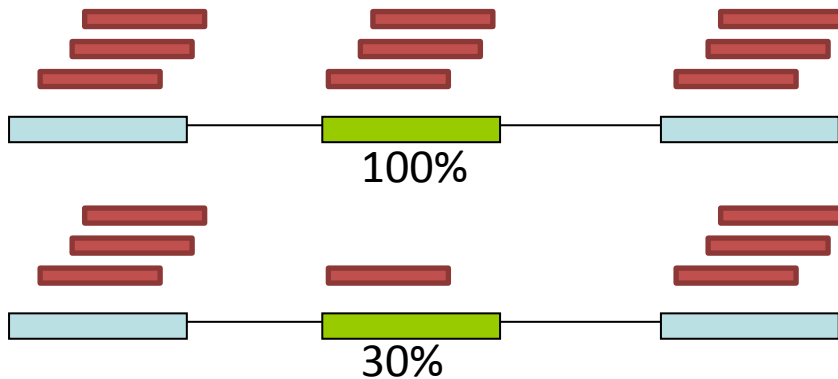
Mature mRNA



How to estimate exon inclusion?

1. Relative coverage:

$\text{coverage}(\text{alt.exon}) / \text{coverage}(\text{const.exon})$

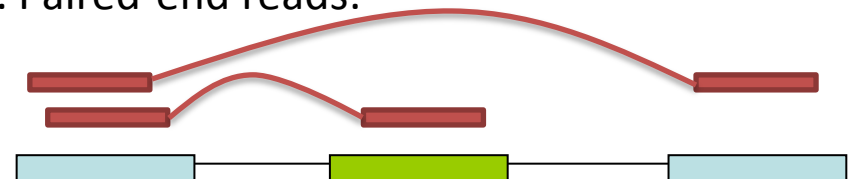


2. Reads covering exon-exon junctions: separate mapping of left and right fragments of reads

inclusion

exclusion

2A. Paired-end reads:



Map reads:
bioscope/bowtie

P-values of differential
splicing: **cuffdiff**