

Содержание

От издательства	16
-----------------------	----

Глава 1. В каком месте генома начинается репликация ДНК?	17
---	----

Путешествие в тысячу миль	18
Скрытые сообщения в точке начала репликации	20
DnaA-боксы	20
Скрытые сообщения в «Золотом жуке»	21
Подсчет слов	22
Задача поиска часто встречающихся слов	23
Более быстрый подход к задаче частых слов	24
Часто используемые слова в <i>Vibrio cholerae</i>	26
Некоторые скрытые сообщения более примечательны, чем другие	27
Взрыв скрытых сообщений	31
Поиск скрытых сообщений в нескольких геномах	31
Задача поиска сгустков	32
Самое простое объяснение процесса репликации ДНК	34
Асимметрия репликации	37
Специфическая статистика прямой и обратной полупепей	41
Неизвестный биологический феномен или статистическая случайность?	41
Дезаминирование	43
Диаграмма смещения	44
Некоторые скрытые сообщения более неуловимы, чем другие	47
Последняя попытка найти <i>DnaA</i> -боксы в <i>E. Coli</i>	51
Эпилог. Осложнения в предсказаниях <i>ori</i>	53
Открытые проблемы	55
Множественные точки начала репликации в бактериальном геноме	55
Поиск источников репликации у архей	57
Поиск точек начала репликации у дрожжей	59
Вычисление вероятностей паттернов в строке	60
Зарядные станции	61
Массив частот	61
Преобразование Patterns в Numbers и наоборот	63
Поиск часто встречающихся слов путем сортировки	65

Решение задачи поиска сгустков	66
Решение задачи часто встречающихся слов с несовпадениями	68
Генерация окрестности строки	69
Поиск часто встречающихся слов с несовпадениями путем сортировки.....	71
Сопутствующие материалы	72
Оценка «О большого» (Big-O).....	72
Вероятности паттернов в строке	73
Самый красивый эксперимент в биологии.....	78
Направленность цепей ДНК.....	80
Ханойские башни.....	81
Парадокс перекрывающихся слов	83
Библиографические примечания.....	85

Глава 2. Какие сегменты ДНК играют роль молекулярных часов?

Есть ли у нас «часовой ген»?	88
Найти мотив сложнее, чем вы думаете	89
Идентификация вечернего элемента	89
Игра в прятки с мотивами.....	90
Метод грубой силы поиска мотива.....	92
Считаем мотивы	93
От мотивов к матрицам профиля и консенсусным строкам	93
На пути к более адекватной функции оценки мотивов.....	96
Энтропия и motif logo	97
От поиска мотива к поиску медианной строки.....	98
Задача поиска мотива.....	98
Переформулировка задачи поиска мотива.....	99
Задача поиска медианной строки.....	101
Почему мы переформулировали задачу поиска мотива?.....	103
Жадный алгоритм поиска мотива.....	104
Использование матрицы профиля для бросания костей.....	104
Анализ жадного алгоритма поиска мотива	106
Поиск мотива и Оливер Кромвель.....	107
Какова вероятность того, что завтра не взойдет солнце?.....	107
Правило преемственности Лапласа.....	108
Улучшенный алгоритм жадного поиска мотивов	109
Рандомизированный поиск мотива.....	112
Игра в кости для поиска мотивов	112
Почему рандомизированный поиск мотивов работает.....	114
Почему рандомизированный алгоритм работает так хорошо?	116
Сэмплирование по Гиббсу	119
Сэмплирование по Гиббсу в действии	121
Эпилог. Как туберкулез впадает в спячку, чтобы спрятаться от антибиотиков?	124
Зарядная станция	127
Решение задачи медианной строки	127
Сопутствующие материалы	128
Экспрессия генов	128
ДНК-чипы	128
Игла Бюффона	129

Сложности в поиске мотива.....	132
Относительная энтропия	132
Библиографические примечания.....	134
Глава 3. Как мы собираем геномы?	135
Взрывающиеся газеты.....	136
Задача реконструкции строки	139
Сборка генома сложнее, чем вы думаете	139
Реконструкция строк из k-меров	139
Повторы усложняют сборку генома.....	142
Реконструкция строк как прогулка по графу перекрытий	143
От строки к графу.....	143
Геном исчезает	146
Два способа представления графов.....	147
Гамильтоновы пути и универсальные строки	148
Другой граф для реконструкции строк	150
Склеивание узлов и графы де Брюйна	150
Прогулка по графу де Брюйна.....	152
Эйлеровы пути	152
Другой способ построения графов де Брюйна.....	153
Построение графов де Брюйна из композиции k-меров	155
Графы де Брюйна в сравнении с графами перекрытия	156
Семь мостов Кенигсберга.....	157
Теорема Эйлера.....	160
От теоремы Эйлера к алгоритму нахождения эйлеровых циклов	163
Построение эйлеровых циклов.....	163
От эйлеровых циклов к эйлеровым путям.....	164
Создание универсальных строк.....	165
Сборка геномов из рид-пар	167
От ридов к рид-парам.....	167
Преобразование рид-пар в длинные виртуальные риды	169
От композиции к спаренной композиции.....	170
Парные графы графы де Брюйна	172
Ловушка парных графов де Брюйна	173
Эпилог. Сборка генома – работа с реальными данными секвенирования.....	176
Разбиваем риды на k-меры	176
Фрагментация генома на контиги.....	177
Сборка ридов с возможными ошибками	179
Определение кратности ребер в графах де Брюйна.....	180
Зарядные станции	181
Влияние склейки на матрицу смежности	181
Генерация всех эйлеровых циклов	182
Реконструкция строки, записанной как путь в парном графе де Брюйна.....	184
Максимальные неветвящиеся пути в графе	186
Сопутствующие материалы	187
Краткая история технологий секвенирования ДНК	187
Повторы в геноме человека	189
Графы	190
Игра «Икосиан»	193
Разрешимые и неразрешимые задачи	194

От Эйлера до Гамильтона и де Брюйна	195
Семь мостов Калининграда	196
Подводные камни сборки двухцепочечной ДНК	197
«ЛУЧШАЯ» теорема	198
Библиографические примечания	199

Глава 4. Как мы секвенируем антибиотики?

Открытие антибиотиков	202
Как бактерии производят антибиотики?	203
Как пептиды кодируются геномом	203
Где в геноме <i>Bacillus brevis</i> закодирован тироцидин?	206
От линейных к циклическим пептидам	207
Уклоняясь от центральной догмы молекулярной биологии	208
Секвенирование антибиотиков путем их дробления на части	209
Введение в масс-спектрометрию	209
Задача секвенирования циклопептидов	210
Алгоритм грубой силы для секвенирования циклопептидов	212
Алгоритм ветвей и границ для секвенирования циклопептидов	214
Масс-спектрометрия и гольф	217
От теоретических к реальным спектрам	217
Адаптация секвенирования циклопептидов для спектров с ошибками	218
От 20 до более чем 100 аминокислот	222
Спектральная свертка спасает положение	223
Эпилог. От смоделированных спектров – к реальным	227
Зарядные станции	229
Создание теоретического спектра пептида	229
Насколько быстро выполняется алгоритм CyclopeptideSequencing?	231
Сокращение списка пептидов Leaderboard	232
Сопутствующие материалы	233
Гаузе и лысенковщина	233
Открытие кодонов	235
Чувство кворума	236
Молекулярная масса	236
Сленоцистеин и пирролизин	237
Псевдополиномиальный алгоритм для Теоремы магистрали	237
Расщепленные гены	238
Библиографические примечания	240

Глава 5. Как мы сравниваем участки ДНК?

Взлом нерибосомного кода	242
Клуб галстуков РНК	242
От сравнения белков к нерибосомному коду	242
Что общего между онкогенами и факторами роста?	244
Введение в выравнивание последовательностей	245
Выравнивание последовательности как игра	245
Выравнивание последовательностей и самая длинная общая подпоследовательность	247
Туристическая задача Манхэттена	248
Какова наилучшая стратегия осмотра достопримечательностей?	248

Достопримечательности в произвольном ориентированном графе	252
Выравнивание последовательности – это замаскированная туристическая задача Манхэттена	253
Введение в динамическое программирование: задача размена монет	257
Жадный обмен денег	257
Рекурсивный обмен денег	258
Размениваем деньги с помощью динамического программирования	259
Новый взгляд на туристическую задачу Манхэттена	261
От Манхэттена к произвольному DAG	266
Выравнивание последовательности как построение графа в стиле Манхэттена	266
Динамическое программирование в произвольном графе DAG	267
Топологические порядки	269
Возвращаясь к графу выравнивания	274
Считаем выравнивания	277
Что не так с моделью LCS?	277
Матрицы счета	279
От глобального к локальному выравниванию	280
Глобальное выравнивание	280
Ограничения глобального выравнивания	281
Бесплатные поездки на такси в графе выравнивания	284
Меняющиеся грани выравнивания последовательности	287
Задача 1. Расстояние редактирования	287
Задача 2. Настройка выравнивания	288
Задача 3. Выравнивание с перекрытием	289
Штрафы за вставки и удаления при выравнивании последовательности	290
Штрафы за аффинные пробелы	290
Строительство графа Манхэттена на трех уровнях	293
Компактное выравнивание последовательности	296
Вычисление счета выравнивания с использованием линейной памяти	296
Задача среднего узла	298
Удивительно быстрый и экономичный алгоритм выравнивания	301
Задача среднего ребра	303
Эпилог. Множественное выравнивание последовательностей	305
Построение трехмерного Манхэттена	305
Жадный алгоритм множественного выравнивания	307
Сопутствующие материалы	310
Светлячки и нерибосомный код	310
Поиск LCS без постройки города	311
Построение топологической сортировки	312
Матрица счета РАМ	313
Алгоритмы «разделяй и властвуй»	314
Счет множественных выравниваний	316
Библиографические примечания	318

Глава 6. Есть ли в человеческом геноме «хрупкие» области?

О мышах и людях	320
Насколько различаются геномы человека и мыши?	321
Синтенные блоки	321

Реверсии	322
Точки перестановки	324
Модель эволюции хромосом со случайными разрывами	325
Сортировка по реверсиям	328
Жадный алгоритм сортировки по реверсиям	332
Точки останова	334
Что такое точки останова?	334
Счет точек останова	335
Сортировка по реверсиям для устранения точек останова	336
Рекомбинации в геномах опухолей	338
От монохромосомных к мультихромосомным геномам	339
Транслокации, слияния и расщепления	339
От генома к графу	340
Двойные разрывы	341
Графы точек останова	344
Вычисление дистанции двойного разрыва	347
Горячие точки рекомбинации в геноме человека	350
Модель случайных разрывов соответствует теореме о дистанции двойного разрыва	350
Модель хрупких разрывов	351
Эпилог. Конструирование синтенных блоков	353
Геномные точечные диаграммы и общие k-меры	353
Поиск общих k-меров	354
Построение синтенных блоков из общих k-меров	357
Синтенные блоки как связанные компоненты в графах	359
Зарядные станции	363
От геномов к графу точек останова	363
Решение задачи сортировки по двойным разрывам	366
Сопутствующие материалы	368
Почему генный состав X-хромосом так консервативен?	368
Открытие геномных рекомбинаций	368
Экспоненциальное распределение	369
Сортировка блинов Билла Гейтса и Дэвида Х. Коэна	370
Сортировка линейных перестановок по реверсиям	371
Библиографические примечания	373

Глава 7. Какое животное заразило нас

коронавирусом?	375
Самая быстрая вспышка	376
Проблемы в отеле «Метрополь»	376
Эволюция SARS	376
Преобразование матриц расстояний в эволюционные деревья	378
Построение матрицы расстояний из геномов коронавируса	378
Эволюционные деревья в виде графов	379
Построение филогении по расстояниям	383
На пути к алгоритму построения филогении по расстоянию	386
В поисках соседних листьев	386
Вычисление длины ветвей	388
Аддитивная филогения	391

Обрезка дерева.....	391
Прикрепление ветви.....	392
Алгоритм реконструкции филогении по расстоянию.....	393
Построение эволюционного дерева коронавирусов	394
Использование метода наименьших квадратов для построения приблизительных филогений.....	395
Ультраметрические эволюционные деревья	397
Алгоритм объединения соседей	402
Преобразование матрицы расстояний в матрицу объединения соседей.....	402
Анализ коронавирусов с помощью алгоритма объединения соседей	406
Ограничения методов реконструкции эволюционного дерева по расстояниям.....	408
Реконструкция эволюционного дерева по признакам	408
Таблицы признаков	408
От анатомических к генетическим признакам	409
Сколько раз эволюция изобретала крылья для насекомых?.....	410
Задача минимального показателя экономии.....	411
Задача максимальной экономии.....	418
Эпилог. Эволюционные деревья в борьбе с преступностью.....	425
Сопутствующие материалы	426
Когда HIV перешел от приматов к человеку?.....	426
Поиск дерева с помощью настройки матрицы расстояний.....	427
Условие четырех точек.....	429
Заразили ли нас атипичной пневмонией летучие мыши?	430
Почему алгоритм объединения соседей работает?	432
Вычисление длин ветвей в алгоритме объединения соседей.....	436
Большая панда: медведь или енот?	437
Откуда пришли люди?	439
Библиографические примечания	441
Глава 8. Как дрожжи научились делать вино?.....	443
Эволюционная история виноделия	444
Как давно мы зависим от алкоголя?	444
Диауксический сдвиг	445
Идентификация генов, ответственных за диауксический сдвиг	445
Две эволюционные гипотезы с разными судьбами	445
Какие гены дрожжей вызывают диауксический сдвиг.....	446
Введение в кластеризацию	447
Анализ экспрессии генов	447
Кластеризация генов дрожжей	451
Принцип правильной кластеризации.....	452
Кластеризация как задача оптимизации.....	454
Самый дальний первый обход.....	456
Самый дальний первый обход	456
Кластеризация k -средних	458
Искажение квадрата ошибки	458
Кластеризация k -средних и центр тяжести	460
Алгоритм Ллойда.....	462
От центров к кластерам и обратно	462
Инициализация алгоритма Ллойда	465

Инициализатор k -means++	466
Кластеризация генов, вовлеченных в диауксический сдвиг	466
Ограничения кластеризации k -средних	468
Ограничения кластеризации k -средних	468
От подбрасывания монеты к кластеризации k -средних	470
Подбрасывание монет с неизвестной симметрией.....	470
В чем же состоит вычислительная задача?	473
От подбрасывания монеты к алгоритму Ллойда	474
Вернемся к кластеризации.....	476
Принятие мягких решений при подбрасывании монет	477
Максимизация ожиданий: E-шаг.....	477
Максимизация ожиданий: M-шаг.....	478
Алгоритм максимизации ожидания.....	480
Мягкая кластеризация k -средних.....	480
Применение алгоритма максимизации ожидания к кластеризации	480
От центров к мягким кластерам	480
От мягких кластеров к центрам.....	483
Иерархическая кластеризация	484
Введение в кластеризацию по расстояниям	484
Определение кластеров по структуре дерева	487
Анализ диауксического сдвига с иерархической кластеризацией.....	490
Эпилог. Кластеризация образцов опухоли	493
Сопутствующие материалы	494
Полногеномная дупликация или серия дупликаций?.....	494
Измерение экспрессии генов	495
ДНК-микрочипы	496
Доказательство теоремы о центре тяжести	496
Матрица экспрессии генов и матрица расстояний/сходств	498
Кластеризация и испорченные клики	499
Библиографические примечания.....	501

Глава 9. Как мы обнаруживаем локацию болезнетворных мутаций?	503
Что вызывает синдром Одо?.....	504
Введение во множественное выравнивание последовательностей	505
Объединение Patterns в префиксное дерево	506
Построение префиксного дерева Trie.....	506
Применение префиксного дерева к множественному выравниванию	508
Предварительная обработка генома как альтернатива	511
Суффиксные попытки (suffix tries)	511
Использование суффиксных попыток для сопоставления последовательностей.....	512
Суффиксные деревья (suffix trees)	514
Суффиксные массивы.....	518
Выравнивание паттерна с суффиксным массивом	519
Преобразование Барроуза–Уилера.....	521
Сжатие генома.....	521
Построение преобразования Барроуза–Уилера.....	521
От повторов к сериям	523
Первая попытка инвертирования преобразования Барроуза–Уилера.....	524

Свойство «первый–последний» и инвертирование преобразования Барроуза–Уилера	527
Свойство «первый–последний»	527
Использование свойства «первый–последний» для инвертирования преобразования Барроуза–Уилера	530
Сопоставление последовательностей с помощью преобразования Барроуза–Уилера	534
Первая попытка сопоставления паттернов Барроуза–Уилера	534
Перемещение по последовательности назад	535
Маппинг «последний–первый»	537
Делаем сопоставление паттернов по Барроузу–Уилеру быстрее	539
Замена маппинга «последний–первый» оценочными массивами	539
Удаление первого столбца матрицы Барроуза–Уилера	542
Где находятся совпадающие паттерны?	543
Барроуз и Уиллер устанавливают контрольные точки	545
Эпилог. Устойчивое к несовпадениям картирование рида	547
Сведение приблизительного сопоставления с паттерном к точному	547
BLAST: Сравнение последовательности с базой данных	549
Приблизительное сопоставление последовательностей с помощью преобразования Барроуза–Уилера	550
Сопутствующие материалы	552
Построение суффиксного дерева	552
Решение задачи самой длинной общей подстроки	555
Построение частичного суффиксного массива	557
Эталонный геном человека	558
Рекомбинации, вставки и делеции в геномах человека	558
Алгоритм Ахо–Корасик	559
Суффиксные массивы и суффиксные деревья	560
Бинарный поиск	565
Библиографические примечания	566

Глава 10. Почему биологи до сих пор не разработали вакцину от ВИЧ?

Классификация фенотипа ВИЧ	567
Каким образом ВИЧ ускользает от иммунной системы человека?	568
Ограничения метода выравнивания последовательностей	570
Азартные игры с якудза	572
Две монеты в рукаве у дилера	573
Поиск CG-островов	575
Скрытые марковские модели	576
От подбрасывания монеты к скрытой марковской модели	576
Диаграмма НММ	577
Переформулировка задачи казино	578
Задача декодирования	581
Граф Витерби	581
Алгоритм Витерби	583
Насколько быстр алгоритм Витерби?	584
Поиск наиболее вероятного результата НММ	586
Профильные НММ для выравнивания последовательностей	588

Как НММ связаны с выравниванием последовательностей?	588
Создание профильной НММ	591
Вероятности перехода и эмиссии профильной НММ	594
Классификация белков с помощью профильных НММ	597
Выравнивание белков по профильной НММ	597
Возвращение псевдосчетов	598
Проблема с молчащими состояниями	601
Действительно ли профильные НММ так полезны?	606
Обучение параметров НММ	608
Определение параметров НММ, когда скрытый путь известен	608
Обучение Витерби	609
Мягкие решения для определения параметров	611
Задача мягкого декодирования	611
Алгоритм «вперед-назад»	612
Обучение Баума-Уэлча	615
Многоликость НММ	617
Эпилог. Природа – мастер, а не изобретатель	618
Сопутствующие материалы	619
Эффект Красной Королевы	619
Гликозилирование	620
Метилирование ДНК	621
Условная вероятность	621
Библиографические примечания	622

Глава 11. Является ли *T. rex* всего лишь гигантской курицей?

Палеонтология встречается с информатикой	624
Какие белки присутствуют в этом образце?	625
Расшифровка идеального спектра	626
От идеального спектра к реальному	629
Секвенирование пептидов	632
Определение пептидов по спектрам	632
Где находятся суффиксные пептиды?	634
Алгоритм секвенирования пептидов	637
Идентификация пептидов	639
Задача идентификации пептидов	639
Идентификация пептидов в неизвестном протеоме <i>T. rex</i>	640
Поиск совпадений пептидов со спектром	640
Идентификация пептидов и теорема о бесконечных обезьянах	642
Частота ложных открытий	642
Статистическая значимость пептид-спектр-совпадений	645
Спектральные словари	647
Пептиды <i>T. rex</i> : постороннее загрязнение или древнее сокровище?	651
Загадка гемоглобина	651
Споры о ДНК динозавров	653
Эпилог. От немодифицированных к модифицированным пептидам. (Часть 1)	654
Посттрансляционные модификации	654
Поиск модификаций как задача выравнивания	655
Построение сетки Манхэттена для спектрального выравнивания	657
Эпилог. От немодифицированных к модифицированным пептидам (Часть 2)	660

Алгоритм спектрального выравнивания	660
Сопутствующие материалы	663
Предсказание генов	663
Поиск всех путей в графе.....	664
Задача антисимметричного пути	665
Преобразование спектров в спектральные векторы.....	666
Теорема о бесконечных обезьянах	670
Вероятностное пространство пептидов в словаре	671
Действительно ли динозавры являются предками птиц?	672
Библиографические примечания	673
Предметный указатель	674

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning Publications очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Глава 1

В каком месте генома начинается репликация ДНК?

Алгоритмическая разминка



Путешествие в тысячу миль...

Репликация генома – одна из важнейших задач, выполняемых в клетке. Прежде чем клетка сможет делиться, она должна сначала реплицировать свой геном, чтобы каждая из двух дочерних клеток наследовала свою собственную копию генома. В 1953 году Джеймс Уотсон и Фрэнсис Крик завершили свою знаменательную статью о двойной спирали ДНК ставшей теперь популярной фразой:

От нашего внимания не ускользнуло, что специфическое выстраивание пар азотистых оснований, которое мы постулировали, сразу предполагает возможный механизм копирования генетического материала.

Они предположили, что две цепи родительской молекулы ДНК раскручиваются во время репликации, и затем каждая родительская цепь действует как матрица для синтеза новой молекулярной цепи. В результате процесс репликации начинается с пары комплементарных цепей ДНК и заканчивается двумя парами комплементарных цепей, как показано на рис. 1.1.

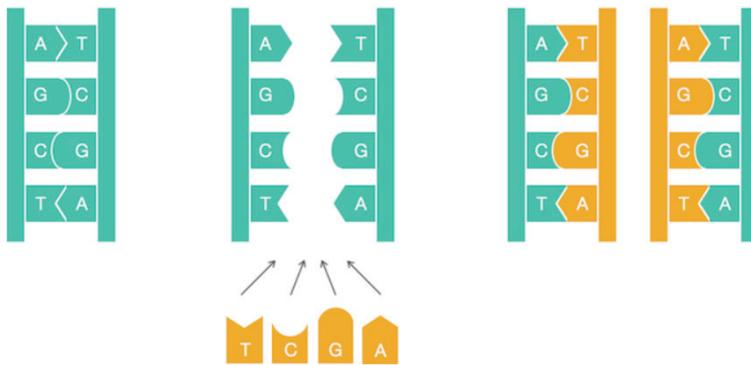


Рис. 1.1 Наивный взгляд на репликацию ДНК. Нуклеотиды аденин (А) и тимин (Т) комплементарны друг другу, как и цитозин (С) с гуанином (G). Комплементарные нуклеотиды связываются друг с другом в ДНК

Хотя на рис. 1.1 репликация ДНК представлена на простом уровне, детали репликации оказались гораздо более сложными, чем предполагали Уотсон и Крик; как мы увидим далее, для обеспечения репликации ДНК требуется поразительный механизм молекулярной логистики.

На первый взгляд, ученый-компьютерщик может и не подумать, что эти детали имеют какое-либо значение для вычислений. Чтобы алгоритмически имитировать процесс, показанный на рис. 1.1, нам нужно всего лишь взять строку, представляющую геном, и выдать ее копию! И все же, если мы найдем время для обзора лежащего в основе этого биологического процесса, мы будем вознаграждены новыми алгоритмическими идеями, полученными в анализе процесса репликации.

Репликация начинается в области генома, называемой **точкой начала репликации** (обозначается *ori*), и выполняется молекулярными копирующими машинами, называемыми **ДНК-полимеразами**. Обнаружение *ori* представляет собой важную задачу не только для понимания того, как клетки реплицируются, но и для решения различных биомедицинских задач. Например, в некоторых методах генной терапии используются генетически сконструированные мини-геномы, которые называются **вирусными векторами**, потому что они способны проникать через клеточные стенки (прямо как настоящие вирусы). Вирусные векторы, несущие искусственные гены, использовались в сельском хозяйстве для создания морозостойчивых томатов и кукурузы, устойчивой к пестицидам. В 1990 году генная терапия была впервые успешно проведена на людях, когда она спасла жизнь четырехлетней девочке, страдающей тяжелым комбинированным иммунодефицитом; девочка была настолько уязвима для инфекций, что была вынуждена жить в стерильной среде.

Идея генной терапии состоит в том, чтобы намеренно заразить пациента, у которого отсутствует важный для жизнедеятельности ген, вирусным вектором, содержащим искусственный ген, кодирующий так называемый терапевтический белок. Оказавшись внутри клетки, вектор реплицируется и в конечном итоге производит множество копий терапевтического белка, который, в свою очередь, лечит болезнь пациента. Чтобы гарантировать, что вектор действительно реплицируется внутри клетки, биологи должны знать, где находится точка начала репликации, *ori*, в геноме вектора, и убедиться, что выполняемые ими генетические манипуляции не влияют на него.

В следующей задаче мы предполагаем, что геном имеет одно начало репликации и представлен в виде **цепи ДНК** или цепи нуклеотидов из четырехбуквенного алфавита {A, C, G, T}.

Задача поиска точки начала репликации: *найти ori в геноме*

Input: ДНК-цепь *Genome*.

Output: локация *ori* в *Genome*.



ОСТАНОВИТЕСЬ и задумайтесь. Является ли эта биологическая задача четко сформулированной вычислительной задачей?

Хотя задача поиска точки начала репликации ставит законный биологический вопрос, она не представляет собой четко сформулированную вычислительную задачу. Действительно, биологи могут немедленно запланировать эксперимент по обнаружению *ori*: например, они могут удалять различные короткие сегменты генома, пытаясь найти сегмент, удаление которого останавливает репликацию. Но специалисты по информатике, с другой стороны,

покачали бы головами и потребовали бы больше информации, прежде чем они смогли бы даже начать думать о задаче.

Почему биологов должно волновать, что думают ученые-компьютерщики? Вычислительные методы в настоящее время являются единственным реальным способом ответить на многие вопросы современной биологии. Во-первых, эти методы намного быстрее, чем экспериментальные методы; во-вторых, результаты многих экспериментов не могут быть интерпретированы без вычислительного анализа. В частности, существующие экспериментальные методы определения локации *ori* требуют много времени. В результате *ori* был экспериментально обнаружен только у нескольких видов. Таким образом, мы хотели бы разработать вычислительный метод поиска *ori*, чтобы биологи могли тратить свое время и деньги на другие задачи.

Скрытые сообщения в точке начала репликации

DnaA-боксы

В оставшейся части этой главы мы сосредоточимся на относительно простом случае обнаружения *ori* в бактериальных геномах, большинство из которых состоит из одной кольцевой хромосомы. Исследования показали, что область бактериального генома, кодирующая *ori*, обычно имеет длину в несколько сотен нуклеотидов. Наш план состоит в том, чтобы начать с бактерии, у которой известны ее *ori*, а затем определить, что делает этот участок генома особенным, чтобы разработать вычислительный метод для нахождения *ori* у других бактерий. Наш пример – *Vibrio cholerae*, бактерия, вызывающая холеру; вот последовательность нуклеотидов, расположенная в ее *ori*:

```
Atcaatgatcaacgtaagcttctaagcatgatcaaggtgctcacacagtttat
ссасаассctgagtggatgacatcaagataggtcggtgtatctccttctctcg
tactctcatgaccacggaaagatgatcaagagaggatgatttcttgccatat
cgcaatgaataacttgtgacttgtgcttccaattgacatcttcagcgccatatt
gсgctggссaaggtgacggagcgggattacgaaagcatgatcatggctgttgt
tctgtttatcttgttttgactgagacttgttaggatagacggtttttcatcac
tgactagссaaagccttactctgcctgacatcgaccgtaaattgataatgaat
ttacatgcttccgсgacgatttacctcttgatcatcgatccgattgaagatct
tcaattgttaattctcttgcctcgactcatagccatgatgagctcttgatcat
gtttccttaaccctctattttttacggaagaatgatcaagctgctgctcttga
tcatcgtttc
```

[Загрузить данные 1.1](#)

Откуда бактериальная клетка знает, что нужно начинать репликацию именно в этом коротком участке гораздо более длинной хромосомы *Vibrio cholerae*,

состоящей из 1 108 250 нуклеотидов? Должно быть какое-то скрытое сообщение в этой области, приказывающее клетке начать репликацию именно здесь. Действительно, мы знаем, что инициация репликации опосредована *DnaA*, белком, который связывается с коротким сегментом внутри *ori*, известным как ***DnaA*-бокс**. Вы можете думать о *DnaA*-боксе как о сообщении в цепи ДНК, общающемся *DnaA* белку: «Присоединяйся сюда!» Вопрос в том, как найти это скрытое сообщение, не зная заранее, как оно выглядит, – сможете ли вы его найти? Другими словами, можете ли вы найти что-то, чем выделяется *ori*? Это рассуждение ставит следующую задачу.

Задача поиска скрытого сообщения: найти скрытое сообщение в точке начала репликации.

Input: строка *Text* (представляющая начало репликации генома).

Output: скрытое сообщение в *Text*.



ОСТАНОВИТЕСЬ и задумайтесь. Представляет ли эта задача четко сформулированную вычислительную задачу?

Скрытые сообщения в «Золотом жуке»

Хотя задача скрытого сообщения ставит законный интуитивный вопрос, она все еще не имеет абсолютно никакого смысла для ученого-компьютерщика, поскольку понятие скрытого сообщения точно не определено. Область *ori* холерного вибриона в настоящее время так же загадочна, как пергамент, обнаруженный Уильямом Леграном в рассказе Эдгара Аллана По «Золотой жук». На пергаменте было написано следующее:

53++!305))6*;4826)4+.)4+);806*;48!8'60))85;1+(;+*8!83(88)5*!;46(;88*96*?;
8)*+;(485);5*!2.*+(;4956*2(5*4)8'8*;4069285);6!8)4++;1(+9;48081;8:8+1;48!
85:4)485!528806*81(+9;48;(88;4(+?34;48)4+;161;:188;+?;

Увидев пергамент, рассказчик замечает: «Если бы все драгоценности Голконды ждали меня после решения этой загадки, я совершенно уверен, что не смог бы их заработать». Легран возражает: «Вполне можно сомневаться, способна ли человеческая изобретательность построить такого рода загадку, которую человеческая находчивость при правильном применении не могла бы разрешить». Он обращает внимание, что три последовательных символа «;48» появляются на пергаменте с удивительной частотой:

53++!305))6;**48**26)4+.)4+);806*;**48**!8'60))85;1+(;+*8!83(88)5*!;46(;88*96*?;8
)*(;**48**5);5*!2.*+(;4956*2(5*4)8'8*;4069285);6!8)4++;1(+9;**48**081;8:8+1;**48**!
85:4)485!528806*81(+9 ;**48**;(88;4(+?34;**48**)4+;161;:188;+?;

Легран ранее уже сделал вывод, что пираты говорят по-английски; поэтому он предположил, что высокая частота «;48» подразумевает, что она кодирует наиболее часто встречающееся английское слово – артикль «THE». Заменяя каждый символ, Легран получил немного более простой текст для расшифровки, который в конечном итоге привел его к зарытому сокровищу. Можете расшифровать и это сообщение?

53++!305))6*THE26)H+.)H+)TE06*THE!E'60))E5T1+(T:+*E!E3(EE)5*!T
H6(TEE*96*?TE)*+(THE5)T5*!2:*(TH956*2(5*N)E'E*TH0692E5)T)6!E)
H++T1(+9THE0E1TE:E+1THE!E5TH)HE5!52EE06*E1(+9THET(EETH(+?3HT
HE)H+T161T:1EET+?T

Подсчет слов

Опираясь на предположение, что ДНК – это язык сам по себе, давайте воспользуемся методом Леграна и посмотрим, сможем ли мы найти какие-нибудь неожиданно часто встречающиеся «слова» в *ori* холерного вибриона (*Vibrio cholerae*). Имеет смысл искать часто встречающиеся слова в *ori*, потому что для различных биологических процессов определенные цепи нуклеотидов удивительно часто появляются в небольших областях генома. Это связано с тем, что некоторые белки могут связываться с ДНК только в том случае, если присутствует определенная последовательность нуклеотидов, и если существует больше вхождений этой последовательности, то более вероятно, что связывание произойдет успешно. (Также менее вероятно, что мутация нарушит процесс связывания.)

Например, **АСТАТ** – удивительно частая подстрока

ACA**АСТАТ**GCA**ТАСТАТ**CGGGA**АСТАТ**CC**Т**.

Мы используем термин «**k-меры**» для обозначения строки длины k и определяем $Count(Text, Pattern)$ как количество раз, когда k -мер $Pattern$ появляется как подстрока строки $Text$. Следуя приведенному выше примеру,

$Count(ACA**АСТАТ**GCA**ТАСТАТ**CGGGA**АСТАТ**CC**Т**, **АСТАТ**) = 3$.

Обратите внимание, что $Count(CGATATATCCATAG, ATA)$ равно 3 (а не 2), так как мы должны учитывать перекрывающиеся вхождения $Pattern$ в $Text$.

Наш план вычисления $Count(Text, Pattern)$ состоит в том, чтобы «сдвигать окно» вдоль строки $Text$, проверяя, соответствует ли каждая подстрока k -мера текста $Text$ образцу $Pattern$. Поэтому мы будем ссылаться на k -мер, начинающийся с позиции i текста, как на $Text(i, k)$. В этой книге мы часто будем использовать **ноль-индексацию** (нумерацию на основе нуля), что означает, что мы начинаем отсчет с 0, а не с 1. В этом случае $Text$ начинается с позиции 0 и заканчивается в позиции $|Text| - 1$ ($|Text|$ обозначает количество символов в тексте). Например, если $Text = GACCATACTG$, то $Text(4, 3) = ATA$. Обратите внимание, что последний k -мер $Text$ начинается с позиции $|Text| - k$, например последний

3-мер GACCATACTG начинается с позиции $10 - 3 = 7$. Это рассуждение приводит к следующему псевдокоду для вычисления $Count(Text, Pattern)$.

```

PatternCount(Text, Pattern)
  count ← 0
  for  $i \leftarrow 0$  до  $|Text| - |Pattern|$ 
    if  $Text(i, |Pattern|) = Pattern$ 
      count ← count + 1
  return count

```

Важное примечание. В этом тексте мы используем термин **псевдокод** для описания алгоритмов, с которыми сталкиваемся при решении задач современной биологии. Псевдокод – это универсальный метод описания алгоритмов, более точный, чем человеческий язык, но не требующий от нас увязнуть в синтаксисе конкретного языка программирования.

Задача поиска часто встречающихся слов

Мы говорим, что $Pattern$ является наиболее частым k -мером в $Text$, если он максимизирует $Count(Text, Pattern)$ среди всех k -меров. Вы можете видеть, что **АСТАТ** является наиболее частым 5-мером для $Text = \text{ACAАСТАТGCАТАСТАТCGGGAACTATCCT}$, а **АТА** – наиболее частым 3-мером для $Text = \text{CGATATATCCATAG}$.



ОСТАНОВИТЕСЬ и задумайтесь. Может ли строка иметь несколько наиболее часто встречающихся k -меров?

Теперь у нас есть строго определенная вычислительная задача.

Задача поиска часто встречающихся слов: найдите наиболее часто встречающиеся k -меры в строке.

Input: строка $Text$ и целое число k .

Output: все наиболее часто встречающиеся k -меры в тексте.

Прямой алгоритм нахождения наиболее часто встречающихся k -меров в строке $Text$ проверяет все k -меры, встречающиеся в этой строке (имеется $|Text| - k + 1$ таких k -меров), а затем вычисляет, сколько раз каждый k -мер появляется в $Text$. Для реализации этого алгоритма, называемого **FrequentWords**, нам потребуется сгенерировать массив $Count$, где $Count(i)$ содержит $Count(Text, Pattern)$ для $Pattern = Text(i, k)$ (рис. 1.2).

<i>Text</i>	A	C	T	G	A	C	T	C	C	C	A	C	C	C	C
<i>Count</i>	2	1	1	1	2	1	1	3	1	1	1	3	3		

Рис. 1.2 Массив *Count* для *Text* = ACTGACTCCCACCCC и $k = 3$. Например, $Count(0) = Count(4) = 2$, потому что ACT (выделен жирным шрифтом) дважды встречается в строке *Text* в позициях 0 и 4

FrequentWords(*Text*, k)

FrequentPatterns ← пустой набор

for $i \leftarrow 0$ до $|Text| - k$

Pattern ← k -мер *Text*(i , k)

$Count(i) \leftarrow$ **PatternCount**(*Text*, *Pattern*)

maxCount ← максимальная величина массива *Count*

for $i \leftarrow 0$ до $|Text| - k$

if $Count(i) = maxCount$

add *Text*(i , k) до *FrequentPatterns*

удалить дубликаты из *FrequentPatterns*

return *FrequentPatterns*



ОСТАНОВИТЕСЬ и задумайтесь. Насколько быстро работает **FrequentWords**?

Хотя **FrequentWords** находит наиболее часто встречающиеся k -меры, он не очень эффективен. Каждый вызов **PatternCount**(*Text*, *Pattern*) проверяет, является ли k -мер *Pattern* в позиции 0 *Text*, позиции 1 *Text* и т. д. Поскольку каждый k -мер требует $|Text| - k + 1$ таких проверок, каждая из которых требует k сравнений, общее количество шагов **PatternCount**(*Text*, *Pattern*) равно $(|Text| - k + 1) \cdot k$. Более того, **FrequentWords** должен вызывать **PatternCount** $|Text| - k + 1$ раз (по одному разу для каждого k -мера *Text*), так что общее количество шагов равно $(|Text| - k + 1) \cdot (|Text| - k + 1) \cdot k$. Чтобы упростить дело, ученые-компьютерщики часто говорят, что время выполнения **FrequentWords** имеет верхнюю границу $|Text|^2 \cdot k$ шагов, и ссылаются на сложность этого алгоритма как $O(|Text|^2 \cdot k)$ (см. **СОПУТСТВУЮЩИЕ МАТЕРИАЛЫ: Big-O («О большое»»)**).

Если $|Text|$ и k малы, как в случае поиска DnaA-боксов в типичных бактериальных *ori*, тогда алгоритм со временем работы $O(|Text|^2 \cdot k)$ вполне приемлем. Но как только мы найдем новое биологическое приложение, требующее от нас решения задачи часто используемых слов для очень длинного текста, мы быстро столкнемся с проблемами.

Более быстрый подход к задаче частых слов

Если бы вам нужно было решить задачу о часто встречающихся словах вручную для небольшого примера, вы, вероятно, сформировали бы таблицу, подобную

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru