

Оглавление

Об авторе	12
О рецензентах	13
От издательства	14
Предисловие	15
Глава 1. Python и окружающее программное обеспечение	20
Установка необходимого базового программного обеспечения с помощью Anaconda.....	21
Подготовка	21
Как это сделать... ..	23
Дополнительно.....	25
Установка необходимого программного обеспечения с помощью Docker.....	26
Подготовка	27
Как это сделать... ..	27
Смотрите также	28
Взаимодействие с R через rpy2	28
Подготовка	28
Как это сделать... ..	29
Дополнительно.....	35
Смотрите также	35
Демонстрация R magic с Jupyter	36
Подготовка	36
Как это сделать... ..	36
Дополнительно.....	37
Смотрите также	38
Глава 2. Знакомство с NumPy, pandas, Arrow и Matplotlib	39
Использование pandas для обработки побочных эффектов вакцин.....	40
Подготовка	40
Как это сделать... ..	40
Дополнительно.....	45
Смотрите также	45
Устранение подводных камней при использовании pandas DataFrames	46
Подготовка	46
Как это сделать... ..	47
Дополнительно.....	49
Уменьшение потребления памяти pandas DataFrames	49
Подготовка	49

Как это сделать.....	49
Смотрите также	52
Ускорение обработки pandas с помощью Apache Arrow	53
Подготовка	53
Как это сделать.....	53
Дополнительно.....	55
NumPy как основа науки о данных и биоинформатики Python.....	56
Подготовка	56
Как это сделать.....	56
Смотрите также	59
Matplotlib как инструмент создания диаграмм.....	59
Подготовка	60
Как это сделать.....	60
Дополнительно.....	66
Смотрите также	67

Глава 3. Секвенирование следующего поколения 68

Доступ в GenBank и перемещение по базам данных NCBI.....	69
Подготовка	70
Как это сделать.....	70
Дополнительно.....	74
Смотрите также	75
Выполнение базового анализа последовательности	75
Подготовка	75
Как это сделать.....	75
Дополнительно.....	77
Смотрите также	77
Работа с современными форматами последовательностей.....	77
Подготовка	78
Как это сделать.....	79
Дополнительно.....	85
Смотрите также	86
Работа с данными выравнивания	86
Подготовка	87
Как это сделать.....	87
Дополнительно.....	93
Смотрите также	93
Извлечение данных из файлов VCF.....	94
Подготовка	94
Как это сделать.....	95
Дополнительно.....	96
Смотрите также	97
Изучение доступности генома и фильтрация данных SNP	97
Подготовка	97
Как это сделать.....	99
Дополнительно.....	109
Смотрите также	109

Обработка данных NGS с помощью HTSeq	110
Подготовка	110
Как это сделать...	111
Дополнительно.....	113
Глава 4. Продвинутый процессинг данных NGS	114
Подготовка массива данных для анализа	114
Подготовка	115
Как это сделать...	115
Использование информации о менделевских ошибках для контроля качества.....	121
Как это сделать...	121
Дополнительно.....	125
Анализ данных с помощью стандартной статистики.....	125
Как это сделать...	126
Дополнительно.....	130
Поиск геномных особенностей из аннотаций секвенирования	131
Как это сделать...	131
Дополнительно.....	133
Мегагеномика с QIIME 2 Python API	133
Подготовка	134
Как это сделать...	135
Дополнительно.....	138
Глава 5. Работа с геномами	139
Технические требования.....	139
Работа с высококачественными референсными геномами	139
Подготовка	140
Как это сделать...	140
Дополнительно.....	145
Смотрите также	145
Работа с референсными геномами низкого качества.....	146
Подготовка	146
Как это сделать...	147
Дополнительно.....	151
Смотрите также	152
Перебор аннотаций генома	152
Подготовка	152
Как это сделать...	152
Дополнительно.....	154
Смотрите также	155
Извлечение генов из референса с помощью аннотаций.....	155
Подготовка	155
Как это сделать...	155
Дополнительно.....	158
Смотрите также	158
Поиск ортологов с помощью Ensembl REST API.....	159
Подготовка	159

Как это сделать.....	159
Дополнительно.....	162
Получение информации об онтологии генов из Ensembl	162
Подготовка	162
Как это сделать.....	163
Дополнительно.....	166
Смотрите также	167
Глава 6. Популяционная генетика	168
Управление наборами данных с помощью PLINK	169
Подготовка	170
Как это сделать.....	171
Дополнительно.....	175
Смотрите также	176
Использование sgkit для генетического анализа популяции с помощью xargau.....	176
Подготовка	176
Как это сделать.....	176
Дополнительно.....	180
Изучение набора данных с помощью sgkit.....	180
Подготовка	180
Как это сделать.....	180
Дополнительно.....	184
Смотрите также	184
Анализ структуры популяции.....	184
Подготовка	184
Как это сделать.....	185
Смотрите также	191
Выполнение PCA.....	191
Подготовка	191
Как это сделать.....	192
Дополнительно.....	194
Смотрите также	194
Исследование структуры популяции с admixture.....	194
Подготовка	194
Как это сделать.....	195
Дополнительно.....	199
Глава 7. Филогенетика	200
Подготовка набора данных для филогенетического анализа	200
Подготовка	200
Как это сделать.....	201
Дополнительно.....	206
Смотрите также	206
Выравнивание генетических и геномных данных.....	207
Подготовка	207
Как это сделать.....	207

Сравнение последовательностей	209
Подготовка	209
Как это сделать...	209
Дополнительно.....	214
Реконструкция филогенетических деревьев	214
Подготовка	214
Как это сделать...	215
Дополнительно.....	218
Рекурсивная игра с деревьями	219
Подготовка	219
Как это сделать...	219
Дополнительно.....	224
Визуализация филогенетических данных	224
Подготовка	224
Как это сделать...	224
Дополнительно.....	229
Глава 8. Использование Protein Data Bank	230
Поиск белка во множественных базах данных.....	231
Подготовка	231
Как это сделать...	231
Дополнительно	235
Представляем Bio.PDB.....	235
Подготовка	236
Как это сделать...	236
Дополнительно	240
Извлечение дополнительной информации из файла PDB	240
Подготовка	240
Как это сделать...	240
Вычисление молекулярных расстояний в файле PDB.....	244
Подготовка	244
Как это сделать...	245
Выполнение геометрических операций	249
Подготовка	249
Как это сделать...	249
Дополнительно	252
Анимация с PyMOL.....	252
Подготовка	252
Как это сделать...	253
Дополнительно	258
Парсинг файлов mmCIF с помощью Biopython	258
Подготовка	259
Как это сделать...	259
Дополнительно	260
Глава 9. Конвейеры биоинформатики.....	261
Представляем серверы Galaxy	262

Подготовка	262
Как это сделать...	262
Дополнительно	264
Доступ к Galaxy с помощью API.....	264
Подготовка	264
Как это сделать...	266
Развертывание конвейера анализа вариантов с помощью Snakemake	272
Подготовка	272
Как это сделать...	273
Дополнительно	277
Развертывание конвейера анализа вариантов с помощью Nextflow.....	278
Подготовка	279
Как это сделать...	279
Дополнительно	283
Глава 10. Машинное обучение в биоинформатике.....	284
Знакомство со scikit-learn на примере PCA	285
Подготовка	285
Как это сделать...	285
Дополнительно.....	287
Использование кластеризации по PCA для классификации образцов.....	287
Подготовка	288
Как это сделать...	288
Дополнительно.....	293
Изучение признаков рака молочной железы с помощью деревьев принятых решений.....	293
Подготовка	294
Как это сделать...	294
Прогнозирование диагностики рака молочной железы с использованием методов случайного леса	297
Подготовка	297
Как это сделать...	297
Дополнительно.....	299
Глава 11. Параллельная обработка с Dask и Zarr	300
Чтение геномных данных с помощью Zarr.....	301
Подготовка	301
Как это сделать...	301
Дополнительно.....	306
Смотрите также	306
Параллельная обработка данных с использованием многопроцессорности Python.....	306
Подготовка	307
Как это сделать...	307
Дополнительно.....	308
Смотрите также	309

Использование Dask для обработки геномных данных на основе массивов NumPy.....	309
Подготовка	309
Как это сделать... ..	310
Дополнительно.....	314
Смотрите также	314
Планирование задач с помощью dask.distributed	314
Подготовка	314
Как это сделать... ..	316
Дополнительно.....	319
Смотрите также	319

Глава 12. Функциональное программирование в биоинформатике.....320

Представление чистых функций	321
Подготовка	321
Как это сделать... ..	321
Дополнительно.....	323
Представление о немутабельности	323
Подготовка	324
Как это сделать... ..	324
Дополнительно.....	325
Избегание мутабельности для надежности шаблона проектирования.....	325
Подготовка	326
Как это сделать... ..	326
Дополнительно.....	327
Использование ленивого программирования для конвейерной обработки	328
Подготовка	328
Как это сделать... ..	328
Дополнительно.....	330
Ограничения рекурсии в Python	330
Подготовка	331
Как это сделать... ..	331
Дополнительно.....	333
Демонстрация модуля Python functools.....	333
Подготовка	333
Как это сделать... ..	333
Дополнительно.....	335
Смотрите также... ..	335

Предметный указатель336

Об авторе

Тиаго Антао – биоинформатик, который в настоящее время работает в области геномики. Бывший ученый, работавший в области информатики, Тиаго перешел в вычислительную биологию, получив степень магистра биоинформатики на факультете естественных наук Университета Порту, Португалия, и докторскую степень за работу по теме «Распространение лекарственно-устойчивой малярии» в Ливерпульской школе тропической медицины, Великобритания. После защиты докторской диссертации Тиаго работал с наборами данных по человеку в Кембриджском университете, Великобритания, и с данными полногеномного секвенирования комаров в Оксфордском университете, Великобритания, а затем помог создать инфраструктуру биоинформатики в Университете Монтаны, США. В настоящее время он работает дата-инженером в области биотехнологии в Бостоне, штат Массачусетс. Он является одним из соавторов Biopython, основного программного пакета биоинформатики, написанного на Python.

О рецензентах

Урминдер Сингх – биоинформатик, ученый в области информатики и разработчик нескольких инструментов биоинформатики с открытым исходным кодом. Его образование включает степени в области физики, информатики и вычислительной биологии, в том числе степень доктора наук по биоинформатике Университета штата Айова, США.

Его разнообразные исследовательские интересы включают эволюцию новых генов, точную медицину, социогеномику, машинное обучение в медицине и разработку инструментов и алгоритмов для больших гетерогенных данных. Вы можете посетить его онлайн-страницу по адресу urmi-21.github.io.

Тиффани Хо работает сотрудником по биоинформатике в Embark Veterinary. Она имеет степень бакалавра Калифорнийского университета в Дэвисе в области генетики и геномики, а также степень магистра в области селекции растений и генетики Корнельского университета.

Предисловие

Биоинформатика – это активная область исследований, в которой используется ряд простых и сложных вычислений для извлечения ценной информации из биологических данных, и эта книга покажет вам, как управлять этими задачами с помощью Python.

Это обновленное издание книги рецептов по биоинформатике с Python начинается с краткого обзора различных инструментов и библиотек в программном окружении Python, которые помогут вам преобразовывать, анализировать и визуализировать наборы биологических данных. По мере продвижения по главам вы познакомитесь с важнейшими методами секвенирования следующего поколения, анализа отдельных клеток, геномики, метагеномики, популяционной генетики, филогенетики и протеомики на основе реальных примеров. Вы узнаете, как работать с лучшими конвейерными системами, такими как серверы Galaxy и Snakemake, освоите различные модули Python для функционального и асинхронного программирования. В числе прочего эта книга поможет вам изучить такие темы, как обнаружение SNP с использованием статистических подходов в высокопроизводительных вычислительных средах, включая Dask и Spark, и применение алгоритмов машинного обучения в биоинформатике.

К концу этой книги по биоинформатике Python вы приобретете знания для реализации новейших методов и сред программирования, что позволит вам работать с данными биоинформатики на любом уровне.

Для кого эта книга

Эта книга предназначена для аналитиков в области биоинформатики, специалистов по данным, вычислительных биологов, исследователей, разработчиков Python, которые хотят решать проблемы биологии и биоинформатики от среднего до высокого уровня. Приветствуется практическое знание языка программирования Python. Также не помешали бы базовые знания по биологии.

Какие темы охватывает эта книга

В главе 1 «*Python и окружающее программное обеспечение*» рассказывается, как создать современную биоинформационную среду с помощью Python. В этой главе обсуждается, как развертывать программное обеспечение с помощью Docker, взаимодействовать с R и с Jupyter Notebooks.

В главе 2 «*Знакомство с NumPy, pandas, Arrow и Matplotlib*» представлены основные библиотеки Python для обработки данных: NumPy для обработки массивов и матриц; Pandas для работы с табличными данными; Arrow для оптимизации обработки Pandas и Matplotlib для построения диаграмм.

В главе 3 «*Секвенирование следующего поколения*» представлены конкретные решения для работы с данными секвенирования на уровне следующего поколения. В этой главе вы узнаете, как работать с большими файлами FASTQ, BAM и VCF. Также обсуждается фильтрация данных.

В главе 4 «*Продвинутый процессинг данных NGS*» рассматриваются расширенные методы программирования для фильтрации данных NGS. Они включают в себя использование менделевских наборов данных, которые затем анализируются стандартной статистикой. Также мы вводим метагеномный анализ.

В главе 5 «*Работа с геномами*» рассматриваются не только высококачественные образцы, такие как геном человека, но также обсуждается, как анализировать другие, низкокачественные образцы, характерные для немодельных видов. Эта глава знакомит вас с обработкой GFF, учит анализировать информацию о геномных особенностях и обсуждает, как использовать генные онтологии.

В главе 6 «*Популяционная генетика*» описывается, как проводить популяционно-генетический анализ эмпирических наборов данных. Например, в Python мы можем выполнять анализ основных компонентов (PCA), компьютерный индекс фиксации (FST) или диаграммы примесных структур.

В главе 7 «*Филогенетика*» используются полные последовательности недавно секвенированных вирусов Эбола для проведения полноценного филогенетического анализа, который включает в себя реконструкцию филогенетического дерева и сравнение последовательностей. В этой главе обсуждаются рекурсивные алгоритмы для обработки древовидных структур.

Глава 8 «*Использование Protein Data Bank*» посвящена обработке файлов банка данных о белках (PDB), например выполнению геометрического анализа структуры белков. В этой главе рассматривается визуализация структуры белков.

В главе 9 «*Конвейеры биоинформатики*» представлены два типа конвейеров обработки. Первый тип конвейера – это Galaxy на основе Python, широко используемая система с веб-интерфейсом, ориентированная в основном на пользователей, не занимающихся программированием, хотя биоинформатикам, возможно, все же придется взаимодействовать с ней программно. Второй тип основан на snakemake и nextflow, типе конвейера, предназначенном для программистов.

Глава 10 «*Машинное обучение в биоинформатике*» знакомит с машинным обучением, использующим интуитивный подход к решению задач вычислительной биологии. В этой главе рассматриваются анализ основных компонентов, кластеризация, деревья принятия решений (Decision Trees) и случайные леса (Random Forests).

Глава 11 «*Параллельная обработка с помощью Dask и Zarr*» познакомит вас с методами работы с очень большими наборами данных и алгоритмами, требующими больших вычислительных ресурсов. В этой главе объясняется, как использовать параллельные вычисления на многих компьютерах (в кластере или облаке). Мы также обсудим эффективное хранение биологических данных.

Глава 12 «*Функциональное программирование для биоинформатики*» познакомит вас с функциональным программированием. Оно позволяет разрабатывать более сложные программы на Python, которые благодаря «ленивому программированию» и немутабельности легче развертывать в параллельных средах со сложными алгоритмами.

ЧТОБЫ ПОЛУЧИТЬ МАКСИМАЛЬНУЮ ОТДАЧУ ОТ ЭТОЙ КНИГИ

Программное и аппаратное обеспечение, описанное в книге	Требования к ОС
Python 3.9	Windows, Mac OS X и Linux (предпочтительно)
Numpy, Pandas, Matplotlib	
Biopython	
Dask, zarr, scikit-learn	

Если вы используете цифровую версию этой книги, мы советуем вам ввести код самостоятельно или получить доступ к коду через репозиторий GitHub (ссылка доступна в следующем разделе). Это поможет вам избежать возможных ошибок, связанных с копированием и вставкой кода.

Загрузите файлы примеров кода

Вы можете загрузить примеры файлов кода для этой книги с GitHub по адресу <https://github.com/PacktPublishing/Bioinformatics-with-Python-Cookbook-Third-edition>. В случае обновления кода он будет обновлен в существующем репозитории GitHub.

У нас также есть другие пакеты кода из нашего богатого каталога книг и видео, доступных по адресу <https://github.com/PacktPublishing/>. Ознакомьтесь с ними!

ЗАГРУЗИТЕ ЦВЕТНЫЕ ИЗОБРАЖЕНИЯ

Мы также предоставляем PDF-файл с цветными изображениями скриншотов и диаграмм, использованных в этой книге. Скачать его можно здесь: <https://packt.link/3KQQO>.

ИСПОЛЬЗУЕМЫЕ СОГЛАШЕНИЯ

В этой книге используется ряд текстовых соглашений.

Код в тексте: указывает кодовые слова в тексте, имена таблиц базы данных, имена папок, имена файлов, расширения файлов, пути, фиктивные URL-адреса, пользовательский ввод и дескрипторы Twitter. Вот пример: «call_genotype имеет вид 56,241×1,1198,2, то есть размерные варианты, образцы, ploidy».

Блок кода устанавливается следующим образом:

```
from Bio import SeqIO
genome_name = 'PlasmoDB-9.3_Pfalciparum3D7_Genome.fasta'
recs = SeqIO.parse(genome_name, 'fasta')
for rec in recs:
    print(rec.description)
```

Когда мы хотим привлечь ваше внимание к определенной части блока кода, соответствующие строки или элементы выделяются жирным шрифтом:

```
AgamP4_2L | organism=Anopheles_gambiae_PEST | version=AgamP4 | length=49364325  
| S0=chromosome
```

```
AgamP4_2R | organism=Anopheles_gambiae_PEST | version=AgamP4 | length=61545105  
| S0=chromosome
```

Жирный: отмечает новый термин, важное слово или слова, которые вы видите на экране. Например, слова в меню или диалоговых окнах отображаются в тексте следующим образом. Вот пример: «По столбцу **Chunk** см. главу 11, но пока вы можете спокойно это игнорировать».

Советы или важные примечания

Представлены так.

РАЗДЕЛЫ

В этой книге вы найдете несколько часто встречающихся заголовков (*Подготовка, Как это сделать..., Как это работает..., Дополнительно и См. также*).

Чтобы получить четкие инструкции по завершению рецепта, используйте эти разделы следующим образом:

Подготовка

В этом разделе рассказывается, чего ожидать от рецепта, и описывается, как установить нужное программное обеспечение или сделать предварительные настройки, необходимые для выполнения рецепта.

Как это сделать...

Этот раздел содержит шаги, необходимые для выполнения рецепта.

Как это работает...

Этот раздел обычно состоит из подробного объяснения того, что произошло в предыдущем разделе.

Дополнительно...

Этот раздел содержит дополнительную информацию о рецепте, чтобы вы могли больше о нем узнать.

Смотрите также

Этот раздел содержит полезные ссылки на другую полезную информацию по рецепту.

Python и окружающее программное обеспечение

Мы начнем с установки основного программного обеспечения, необходимого для большей части этой книги. Оно включает в себя дистрибутив **Python**, некоторые фундаментальные библиотеки Python и внешнее программное обеспечение для биоинформатики. Мы также рассмотрим мир за пределами Python. В биоинформатике и больших данных важную роль играет **R**; поэтому вы узнаете, как взаимодействовать с ним через **rpy2**, который представляет собой мост от Python к R. Кроме того, мы изучим преимущества, которые может дать нам фреймворк **IPython** (через Jupyter Lab) для эффективного взаимодействия с R. Учитывая, что управление исходным кодом с помощью **Git** и **GitHub** широко распространено, мы позаботимся о том, чтобы наш сетап хорошо работал с ними. Эта глава даст вам основу для всех процессов вычислительной биологии, которые мы будем выполнять в оставшейся части книги.

Поскольку у разных пользователей разные требования, мы рассмотрим два разных подхода к установке программного обеспечения. Один подход заключается в использовании дистрибутива **Anaconda** Python (<http://docs.continuum.io/anaconda/>), а другой – в установке программного обеспечения через **Docker** (это метод виртуализации сервера, основанный на контейнерах, использующих одно и то же ядро операционной системы; пожалуйста, обратитесь к <https://www.docker.com/>). Это также установит вам Anaconda, но уже внутри контейнера. Если вы используете операционную систему Windows, мы настоятельно рекомендуем вам рассмотреть возможность поменять операционную систему или использовать Docker с помощью некоторых существующих опций в Windows. В macOS вы можете установить большую часть программного обеспечения непосредственно, хотя Docker также доступен. Обучение с использованием локального дистрибутива (Anaconda или чего-то другого) проще, чем с Docker, но, учитывая, что управление пакетами в Python может быть сложным, образы Docker обеспечивают определенный уровень стабильности.

В этой главе мы рассмотрим следующие рецепты:

- установка необходимого программного обеспечения с Anaconda;
- установка необходимого программного обеспечения с Docker;
- взаимодействие с R через rpy2;
- выполнение R magic с Jupyter.

УСТАНОВКА НЕОБХОДИМОГО БАЗОВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ ANACONDA

Прежде чем мы начнем, нам нужно установить необходимое базовое программное обеспечение. Следующие разделы познакомят вас с программным обеспечением и шагами, необходимыми для его установки. Каждая глава и раздел могут предъявить дополнительные требования – мы укажем их, когда достигнем соответствующего момента в изложении. Альтернативный способ начать работу – использовать рецепт Docker, после чего вас обеспечат всем нужным через контейнер Docker.

Если вы уже используете другой дистрибутив Python, мы настоятельно рекомендуем вам рассмотреть Anaconda, поскольку он стал стандартом де-факто для науки о данных и биоинформатики. Также именно этот дистрибутив позволит вам установить программное обеспечение от **Bioconda** (<https://bioconda.github.io/>).

Подготовка

Python можно запускать поверх разных сред. Например, вы можете использовать Python внутри виртуальной машины Java (Java Virtual Machine, **JVM**) (через **Python** или с .NET через **IronPython**). Однако здесь нас интересует не только Python, но и вся среда программного обеспечения вокруг него. Поэтому мы будем использовать стандартную реализацию **CPython**, так как версии JVM и .NET существуют в основном для взаимодействия с оригинальными библиотеками этих платформ.

Для создания наших кодов мы будем использовать Python 3.10. Если вы начинали с Python и биоинформатики, подойдет любая операционная система. Но здесь нас в основном интересует промежуточное и продвинутое применение. Таким образом, если вы используете Windows и macOS, имейте в виду, что большую часть сложного ресурсозатратного анализа придется выполнять в Linux (вероятно, на Linux high-performance computing (высокопроизводительных вычислениях Linux) или кластере **HPC**). Анализ данных секвенирования следующего поколения (Next-generation sequencing, **NGS**) и сложное машинное обучение в основном выполняются на кластерах Linux.

Если вы работаете в Windows, следует подумать об обновлении до Linux для вашей работы в области биоинформатики, поскольку большинство современных программ для биоинформатики не будут работать в Windows. Обратите внимание, что macOS подойдет почти для всех анализов, если только вы не планируете использовать компьютерный кластер, который, скорее всего, будет основан на Linux.

Если вы работаете в Windows или macOS и у вас нет простого доступа к Linux, не беспокойтесь. Вам на помощь придет современное программное обеспечение виртуализации (такое как **VirtualBox** и Docker), которое позволит установить виртуальную Linux на вашу операционную систему. Если вы работаете с Windows и решили, что хотите перейти на оригинальный язык и не использовать Anaconda, будьте осторожны с выбором библиотек; вы, вероятно, будете в большей безопасности, если установите 32-битную версию для всего (включая сам Python).

Примечание

Если вы работаете в Windows, многие инструменты будут вам недоступны.

Биоинформатика и наука о данных развиваются с головокружительной скоростью; это не реклама, это реальность. При установке программных библиотек выбор версии может причинить проблемы. В зависимости от кода, который у вас есть, он может не работать с некоторыми старыми версиями или, возможно, даже не работать с более новой версией. Будем надеяться, что любой код, который вы используете, покажет правильные соответствия, хотя это и не гарантируется. В этой книге мы установим точные версии всех программных пакетов и позаботимся о том, чтобы код работал с ними. Вполне естественно, что код может нуждаться в согласовании с другими версиями пакета.

Программное обеспечение, разработанное для этой книги, доступно по адресу <https://github.com/PacktPublishing/Bioinformatics-with-Python-Cookbook-Third-edition>. Чтобы получить к нему доступ, вам нужно будет установить Git. Опыт работы с ним имеет свою пользу, потому что с помощью Git разрабатывается множество программных пакетов для научных вычислений.

Прежде чем правильно установить стек Python, вам необходимо установить все внешнее программное обеспечение, которое требуется для Python, с которым вы будете взаимодействовать. Список будет варьироваться от главы к главе, и все программные пакеты будут объяснены в соответствующих главах. К счастью, со времени выхода предыдущих изданий этой книги большая часть программного обеспечения для биоинформатики стала доступна через проект Bioconda; поэтому инсталляция, как правило, не вызовет проблем.

Вам нужно будет установить некоторые компиляторы и библиотеки проектирования, все они бесплатны. В Ubuntu рассмотрите возможность установки пакета `build-essential` (`apt-get install build-essential`), а для macOS подумайте о **Xcode** (<https://developer.apple.com/xcode/>).

На рис. 1.1 вы найдете список наиболее важного программного обеспечения для разработки биоинформатики с помощью Python.

Мы будем использовать `pandas` для обработки большинства табличных данных. Альтернативой может быть использование только стандартного Python. Библиотека `pandas` стала настолько распространенной в науке о данных, что, вероятно, имеет смысл обрабатывать с ее помощью все табличные данные (если они помещаются в память).

Вся наша работа будет производиться внутри проекта Jupyter, а именно Jupyter Lab. Jupyter *де-факто* стал стандартом для написания скриптов интерактивного анализа данных. К сожалению, формат по умолчанию для Jupyter Notebooks основан на JSON. Этот формат трудно читать, трудно сравнивать, и его необходимо экспортировать, чтобы передать в обычный интерпретатор Python. Чтобы устранить эту проблему, расширим Jupyter с помощью `jupyter_text` (https://jupyter_text.readthedocs.io/), что позволит нам сохранять блокноты Jupyter как обычные программы Python.

Имя	Использование	URL-адрес	Цель
Project Jupyter	Все главы	https://jupyter.org/	Интерактивные вычисления
pandas	Все главы	https://pandas.pydata.org/	Обработка данных
NumPy	Все главы	http://www.numpy.org/	Обработка массивов/матриц
SciPy	Все главы	https://www.scipy.org/	Научные вычисления
Biopython	Все главы	https://biopython.org/	Библиотека биоинформатики
seaborn	Все главы	http://seaborn.pydata.org/	Библиотека статистических диаграмм
R	Биоинформатика и статистика	https://www.r-project.org/	Язык статистических вычислений
rpy2	R-подключение	https://rpy2.readthedocs.io	Интерфейс R
PyVCF	NGS	https://pyvcf.readthedocs.io	Обработка VCF
Pysam	NGS	https://github.com/pysam-developers/pysam	Обработка SAM/BAM
HTSeq	NGS/геномы	https://htseq.readthedocs.io	Обработка NGS
DendroPY	Филогенетика	https://dendropy.org/	Филогенетика
PyMol	Протеомика	https://pymol.org	Молекулярная визуализация
scikit-learn	Машинное обучение	http://scikit-learn.org	Библиотека машинного обучения
Cython	Большие данные	http://cython.org/	Высокая производительность
Numba	Большие данные	https://numba.pydata.org/	Высокая производительность
Dask	Большие данные	http://dask.pydata.org	Параллельная обработка

Рис. 1.1. Таблица, показывающая различные программные пакеты, полезные в биоинформатике

Как это сделать...

Для начала ознакомьтесь со следующими шагами:

1. Начните с загрузки дистрибутива Anaconda с <https://www.anaconda.com/products/individual>. Мы будем использовать версию 21.05, хотя вам, вероятно, подойдет и более поздняя, если она вам доступна. Вы можете принять все параметры инсталляции по умолчанию, но можете убедиться,

что двоичные файлы conda находятся в вашем пути (не забудьте открыть новое окно, чтобы можно было обновить путь). Если у вас есть другой дистрибутив Python, будьте осторожны с PYTHONPATH и существующими библиотеками Python. Вероятно, лучше отключить ваш PYTHONPATH. Насколько это возможно, удалите все другие версии Python и его библиотеки, установленные ранее.

2. Займемся библиотеками. Теперь мы создадим новую среду conda под названием bioinformatics_base с biopython=1.70, как показано в следующей команде:

```
conda create -n bioinformatics_base python=3.10
```

3. Давайте активируем среду следующим образом:

```
conda activate bioinformatics_base
```

4. Давайте добавим каналы bioconda и conda-forge в наш список источников:

```
conda config --add channels bioconda
conda config --add channels conda-forge
```

5. Также установим основные пакеты:

```
conda install \
biopython==1.79 \
jupyterlab==3.2.1 \
jupyter-text==1.13 \
matplotlib==3.4.3 \
numpy==1.21.3 \
pandas==1.3.4 \
scipy==1.7.1
```

6. Теперь сохраним нашу среду, чтобы мы могли использовать ее позже для создания новых сред на других машинах или если вам нужно будет очистить базовую среду.

```
conda list -explicit > bioinformatics_base.txt
```

7. Можно даже установить R из conda:

```
conda install rpy2 r-essentials r-gridextra
```

Обратите внимание, что r-essentials устанавливает множество пакетов R, включая ggplot2, который мы будем использовать позже. Кроме того, мы устанавливаем r-gridextra, так как будем использовать его в Notebook.

Дополнительно...

Если вы предпочитаете не использовать Anaconda, вы сможете установить разные библиотеки Python через `pip`, используя любой выбранный вами дистрибутив. Вам, вероятно, понадобится довольно много компиляторов и инструментов сборки¹ – не только компиляторы C, но также C++ и Fortran.

Мы не будем использовать среду, созданную на предыдущих шагах. Вместо этого мы будем использовать ее в качестве базы для клонирования рабочих сред. Это связано с тем, что управление средой с помощью Python – даже с помощью системы пакетов `conda` – может причинить головную боль. Таким образом, мы создадим чистую среду, которую мы никогда не испортим и от которой мы сможем избавиться, если она станет неуправляемой.

Например, представьте, что вы хотите создать среду для машинного обучения с помощью `scikit-learn`. Можно сделать следующее.

1. Создайте клон исходной среды:

```
conda create -n scikit-learn --clone bioinformatics_base
```

2. Добавьте `scikit-learn`:

```
conda activate scikit-learn
conda install scikit-learn
```

Находясь внутри JupyterLab, мы должны открывать наши файлы `jupyter` с помощью блокнота Notebook, а не текстового редактора. Поскольку файлы `jupyter` имеют то же расширение, что и файлы Python – это особенность, а не ошибка, – по умолчанию JupyterLab будет использовать обычный текстовый редактор. Когда мы открываем файл `jupyter`, нам нужно переписать установки по умолчанию. Открыв его, щелкните правой кнопкой мыши и выберите **Notebook**, как показано на рис. 1.2.

Наши файлы `jupyter` не будут сохранять графические данные, и это допустимо для данной книги. Если вы хотите иметь версию с изображениями, это возможно с помощью парных блокнотов. Для получения дополнительной информации посетите страницу Jupytertext (<https://github.com/mwouts/jupytertext>).

Предупреждение

Поскольку наш код предназначен для запуска внутри Jupyter, много раз в этой книге я не буду использовать `print` для вывода содержимого, поскольку последняя строка ячейки будет отображаться автоматически. Если вы не используете блокноты, сделайте `print`.

¹ Сборка включает в себя компиляцию, компоновку и упаковку кода в пригодную для использования или исполняемую форму. – *Прим. ред.*

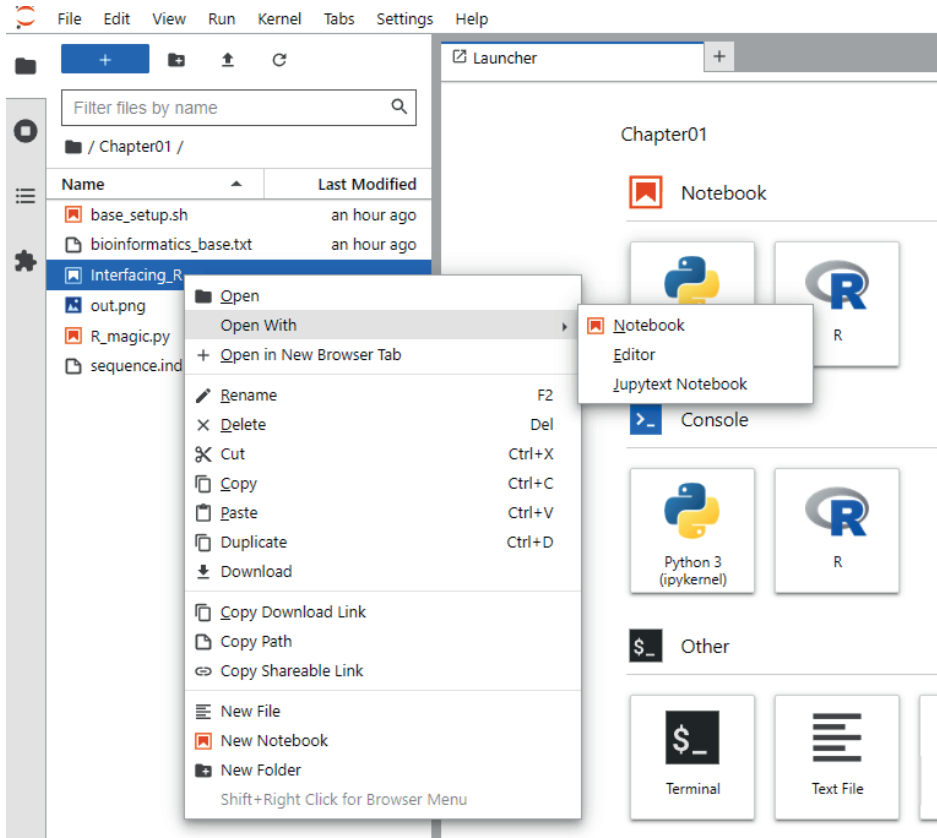


Рис. 1.2. Открытие файла jupyter в Notebook

УСТАНОВКА НЕОБХОДИМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ DOCKER

Docker – наиболее широко используемый фреймворк для реализации виртуализации на уровне операционной системы. Эта технология позволяет вам иметь независимый контейнер: уровень, который легче, чем виртуальная машина, но все же позволяет компартиментализировать программное обеспечение. Он практически изолирует все процессы, создавая ощущение, что каждый контейнер является виртуальной машиной.

Docker хорошо работает на обоих краях разработки: это удобный способ настроить содержание данной книги в учебных целях, и он может стать вашей предпочтительной платформой для развертывания приложений в сложных средах. Этот рецепт является альтернативой предыдущему рецепту.

Однако для сред проектирования, рассчитанных на долгую историю, лучше всего подойдет что-то вроде предыдущего рецепта, хотя это может повлечь за собой более трудоемкую первоначальную настройку.