МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ)"

ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

КАФЕДРА ДИСКРЕТНОЙ МАТЕМАТИКИ

Выпускная квалификационная работа по направлению 01.03.02
ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА
НА ТЕМУ:

Вариационный байесовский вывод в графических моделях в задаче восстановления клональной структуры опухоли

Студент _____ Иванов В.В.
Научный руководитель к.ф-м.н. _____ Юнхуа Хуань.
Зав. кафедрой д.ф-м.н., профессор _____ Райгородский А.М.

МОСКВА, 2020

1 Аннотация

В данной дипломной работе предложена графическая байесовская модель машинного обучения XClone. Её задача — восстановление клонального состава опухоли по данным ДНК-секвенирования одиночных клеток. XClone — статистическая модель опухолевого образца, оптимальные параметры которой подбираются посредством вариационного байесовского вывода. По этим параметрам можно восстановить структурные мутации на каждой из хромосом в клеток образца, что позволяет проследить эволюцию опухоли. В работе описана формальная постановка задачи и приведена реализация алгоритма на языке программирования Python. Актуальность задачи подтверждается тем, что в последние годы несколько статей схожей тематики — SiCloneFit[31], InferCNV[1], Casper[10], CHISEL[29] — было опубликовано в высокоимпактных научных журналах, но среди них не было полных аналогов. Практическую ценность работы подтверждает то, что задача пришла из клинической практики немецких врачей-онкологов. Научная новизна работы заключается в том, что, несмотря на популярность темы, у него пока есть всего один прямой конкурент — алгоритм CHISEL, — от которого XClone выгодно отличается тем, что допускает естественное обобщение на случай нескольких модальностей. Концептуально XClone может поддерживать не только геномные, но и транскриптомные данные, а также информацию о соматических мутациях и митохондриальной ДНК в клетках образца. Каждая из этих модальностей имеет клиническую ценность и позволяет лучше понимать эволюцию опухоли.

[1] https://github.com/broadinstitute/inferCNV/wiki

Содержание

2 Материалы и методы

2.1 Вероятностная модель числа прочтений

При работе с данными секвенирования часто возникает задача оценить матожидание числа прочтений по заданному участку генома. В случае DNA-seq, эта величина описывается простой вероятностной моделью

$$X_i \sim \text{Poisson}(S\, p_i\, Q(g_i)\, m_i)$$

Рис. 2.1: Вероятностая модель числа прочтений X_i в сегменте i в DNA-seq. S — *scale factor*, p_i — число копий сегмента, $Q(g_i)$ — влияние *GC-состав*, m_i — *bin mappability*

Ниже приведены определения этих факторов:

Определение 2.1 (*Scale factor*)**.** Число ридов, которые фрагмент ДНК порождает при секвенировании. Эта величина зависит как от **сложности библиотеки** (матожидание числа различных молекул, которые могут получиться в ходе ПЦР), так и от **глубины секвенирования** (точное значение зависит от технологии, но неформально стоит понимать как число ридов на единицу длины; увеличение амплификации повышает глубину покрытия, но и увеличивает затраты).

Определение 2.2 (*Bin mappability*)**.** Bin mappability неформально следует понимать как долю k-меров из заданного диапазона, которые однозначно выравниваются на этот же диапазон, где k подчиняется Пуассоновской модели данных секвенирования. Если диапазон состоит из повторов одного короткого участка, то его mappability будет низкой, так как однозначно выравниваться будут только риды длиной больше половины от размера этого диапазона, вероятность которых будет мала. При заданной сегментации, эту величину можно с заданной точностью посчитать аналитически, но обычно для этого используют метод Монте-Карло.

Определение 2.3 (*GC-состав*)**.** Доля гуанина (G) и цитозина (C) среди нуклеотидов последовательности. В комплементарной GC-паре три водородных связи вместо двух как у AT-пар, потому последовательности с высоким содержанием G и C более устойчивы к нагреву, а потому реже расщепляются на фрагменты, достаточно короткие для амплификации при ПЦР. Аналогично, если GC-состав очень мал, то велик шанс, что при нагреве последовательность распадётся на слишком маленькие части, к которым уже нельзя будет присоединить праймер. Как следствие, покрытие последовательностей со слишком большим или слишком маленьким GC-составом в среднем ниже.

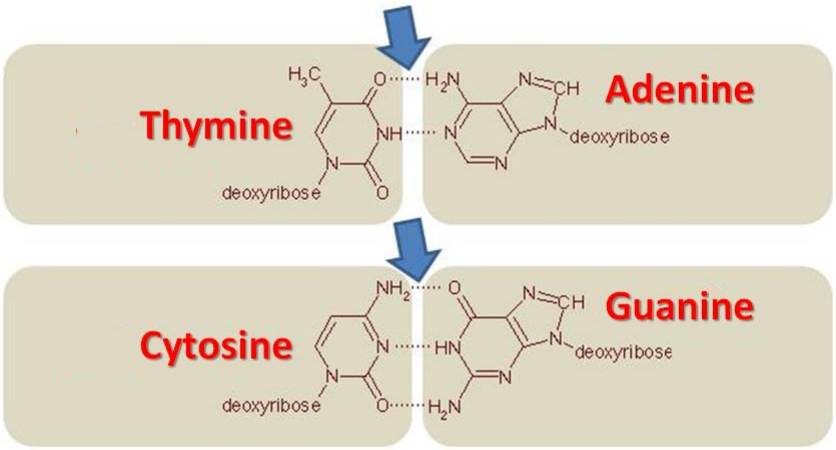


Рис. 2.2: Комлементарные пары: аденин-тимин и гуанин-цитозин

В случае RNA-seq простой модели, к сожалению, быть не может. Дело в том, что геном статичен. Факторов, которые могут повлиять на распределение ридов в DNA-seq, не так много: это либо структурные вариации, либо особенности нуклеотидной последовательности как строки, без привязки к её биологическому смыслу. Картина экспрессии же постоянно меняется. На неё влияет и клеточный цикл, и окружающая среда, и патологии отдельных компонент клетки. Многочисленные регуляторные механизмы не позволяют моделировать экспрессию генов по отдельности: уровни экспрессии часто коррелируют, а иногда и зависят друг от друга нелинейно (т.н. **синергия генов**). Хуже того, в науке хорошо изучено такое явление как **эпистаз** — мутации в одном гене могут

приводить к качественным изменениям фенотипа, выходящим далеко за пределы непосредственных функций этого гена. В современной науке существует множество моделей транскрипции, принимающих во внимание многие из этих факторов, но их содержательный обзор выходит далеко за рамки данной работы.

2.2 Алгоритмы предобработки данных

Профессия вычислительного биолога подразумевает рутинную обработку больших гетерогенных данных, особенно что касается single-cell технологий. В связи с этим был реализован протокол предобработки данных секвенирования, основные шаги которого разобраны в данном разделе.

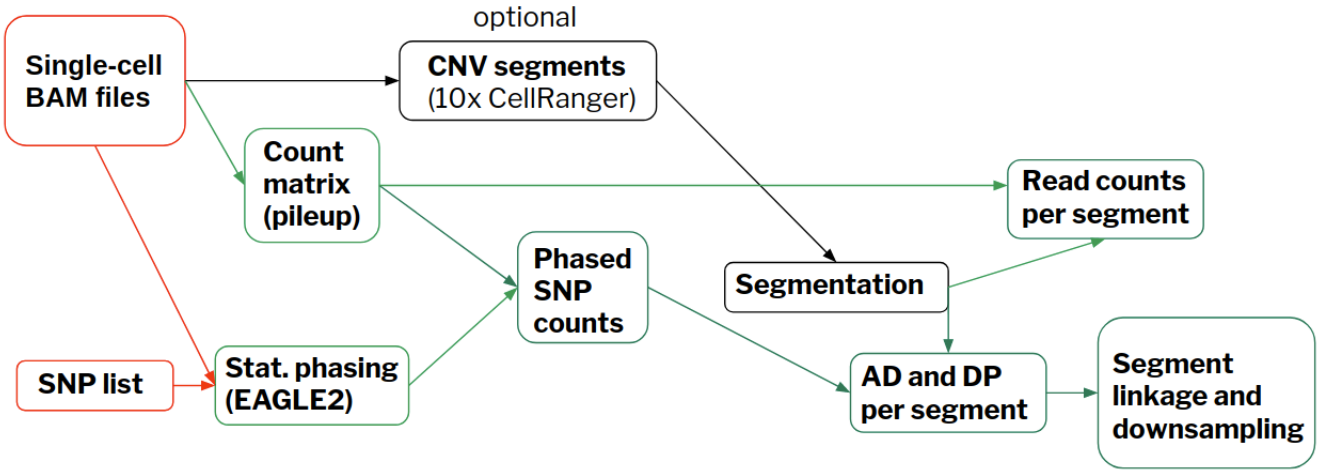


Рис. 2.3: Граф протокола предобработки данных для алгоритма XClone. Красным обозначены входные данные, чёрным — опциональные шаги, зелёным — реализованные стадии.

2.2.1 Извлечение данных из BAM-файлов

BAM — *binary SAM* — *binary sequence alignment/map format* — общепринятый формат сжатого хранения данных секвенирования, с подробностями которого можно ознакомиться в оригинальной публикации [16]. BAM-файл, полученный по протоколам 10x Genomics, занимает до нескольких терабайтов дискового пространства, потому эффективное извлечение ин-

формации из BAM-файлов это нетривиальная инженерная задача. Главные входные файлы XClone — матрицы прочтений. Таких матриц требуется три:

- матрица RD всех прочтений достаточного качества;
- матрица DP всех прочтений, накрывающих хоть один ОНП в пределах сегментов;
- матрица AD всех прочтений, накрывающих хоть альтернативный аллель ОНП в пределах сегментов.

Для получения матрицы RD из данных scRNA-seq был использован протокол **count** из **CellRanger**. Для всех остальных матриц во всех остальных случаях был использован **CellSNP**[2].

2.2.2 Статистическое фазирование гаплотипов

Гаплотипирование — определение того, от какого родителя унаследован каждый алллель в геноме — одна из ключевых задач генетики человека. Сложность её решения обусловлена контекстом, в котором она возникает в современных исследованиях, когда секвенируются порядка $2 \cdot 10^4$ — 10^6 позиций в геномах тысяч человек. Если прочтения короткие и не накрывают много позиций одновременно, то нужно секвенировать обоих родителей каждого участника эксперимента, что непрактично и не всегда возможно. Следовательно, нужно разрабатывать статистические методы гаплотипирования. Они основаны на наблюдении, что некоторые группы аллелей часто наследуются совместно. Это явление называется **неравновесной сцепленностью**. Если прогаплотипировано достаточное количество представителей популяции, то можно построить приближённые таблицы сцепленности и гаплотипировать новые образцы методом максимизации правдоподобия.

На момент написания этого текста, стандартом статистического гаплотипирования считается алгоритм **EAGLE2**[18][3]. Этот алгоритм основан

[2]https://github.com/single-cell-genetics/cellSNP

[3]https://data.broadinstitute.org/alkesgroup/Eagle/

на скрытых марковских моделях и использует 32,470 образца из базы данных **Haplotype Reference Consortium**[5].

Алгоритм EAGLE2 обладает существенным недостатком: его метки имеют только локальный смысл. В пределах окна в 20-50 килобаз любые два ОНП с одинаковой наследуются совместно, но при сдвиге окна смысл меток может спонтанно поменяться на противоположный, это так называемая **ошибка смены цепи**. Т.е. два ОНП с разных концов хромосомы, помеченые одной меткой, могут быть унаследованы от разных родителей. Из-за этого в матрицах прочтений размывается сигнал аллельного дисбаланса: чтобы сделать данные менее разреженными, прочтения соседних небольших сегментов суммируются, в том числе и аллель-специфичные. Ясно, что если среди двух соседних сегментов с одинаковой меткой один полностью унаследован от отца, а второй — от матери, то при сложении их аллель-специфичные сигналы скомпенсируют друг друга. Это, в свою очередь, приводит к неправильному предсказанию аллель-специфичных структурных вариаций и неправильной кластеризации клеток. Авторы EAGLE2 в переписке явно дали понять, что в общем случае детектировать и исправлять такого рода ошибки их подход не позволяет. Но в контексте модели XClone удалось разработать статистический метод, показавший хорошие результаты при устранении ошибок смены цепи. Его подробное описание можно найти в одноимённом разделе.

2.2.3 Подходы к сегментации генома

Одной из основных задач XClone является предсказание **ASCNV** — аллель-специфических структурных вариаций генома. Это происходит в несколько этапов: (1) вначале производится сегментация генома с одновременным подсчётом матриц прочтений, (2) затем глубина покрытия сегментов сравнивается с эталонной для подсчёта RDR, (3) откуда получается оценка общего числа копий, (4) которая затем уточняется при помощи сигналов аллельного дисбаланса. Тем не менее, на точность предсказания влияют ещё и технические факторы, фигурирующие в ве-

роятностной модели числа прочтений. Наиболее существенным фактором является bin mappability.

Подсчёт bin mappability — задача чисто техническая и довольно утомительная, т.к. она подразумевает проведение симуляций процесса секвенирования по какому-то конкретному протоколу. Кроме того, она давно считается решённой, а потому не представляет особого научного интереса. В связи с этим, для отфильтровывания участков низкого качества используется готовое решение — **CellRanger DNA**, алгоритм[4] от 10X Genomics, поставщика оборудования для single-cell севенирования в научной группе автора. Этот алгоритм разбивает геном на сегменты длиной в 20кб, после чего отфильтровывает те, для которых bin mappability меньше, чем 70% (не более 10-15% при использовании референсного генома GRCh37). CellRanger DNA сам по себе является алгоритмом поиска CNV. Тем не менее, он размечает максимально возможную часть генома каждой из клеток, в том числе участки без структурных вариантов. Благодаря этому можно гарантировать, что все участки генома, пригодные для надёжного определения ASCNV, войдут в итоговую сегментацию.

Найденные участки накрывают некоторое подмножество референсного генома, которое затем подразделяется на сегменты размера 20-50 килобаз, в пределах которых вероятность ошибки смены цепи невелика, а потому сигнал аллельного дисбаланса статистически достоверный.

[4]https://support.10xgenomics.com/single-cell-dna/software/pipelines/latest/algorithms/cnv_calling

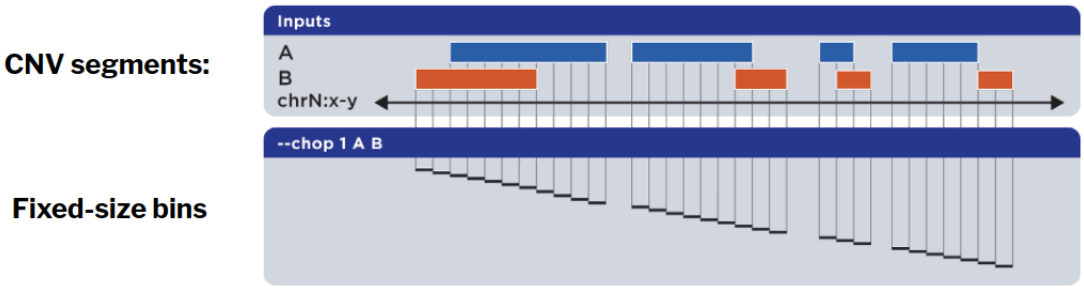


Рис. 2.4: Иллюстрация алгоритма сегментирования генома. Длина индивидуальных сегментов задаётся заранее и выбирается из диапазона 20-50кб. Каждые k подряд идущих фрагментов затем объединяются в блоки. Соответствующие подматрицы прочтений при этом суммируются с одновременной коррекцией ошибок смены цепи.

В силу того, что структурные вариации обычно охватывают участки генома размеров хотя бы в несколько мегабаз, перед началом предсказания уместно аггрегировать подряд идущие сегменты в блоки фиксированного размера (обычно 1-5 мегабаз), чтобы получить менее шумные BAF и RDR. Тем не менее, наивно аггрегировать содержимое сегментов внутри блока — просуммировать числа прочтений — не получится, т.к. можно потерять аллель-специфический сигнал из-за ошибок смены цепи. В связи с этим был разработан алгоритм суммирования с коррекцией ошибок, который разобран в следующем разделе.

Такой подход к сегментации генома используется в заключительной версии XClone. Тем не менее, изначально большие надежды возлагались на более продвинутый метод, основанный на данных секвенирования длинными прочтениями по технологии Oxford Nanopore. Если прочтения достаточно длинные, они могут накрывать сразу несколько ОНП. Благодаря этому их гаплотипы определяются однозначно: просто из нуклеиотидной последовательности рида понятно, какие именно аллели лежат на одной хромосоме. Если покрытие генома достаточно хорошее, то длинные прочтения будут накладываться друг на друга, за счёт чего можно получить достаточно длинные гаплотипы. В силу того, что в первой версии модели ASCNV считались известными, как и клональные

линии клеток в DNA-seq образце, для получения итоговой сегментации диапазоны структурных вариаций, обнаруженных CellRanger DNA, пересекались с гаплотипами, полученными по данным Oxford Nanopore.

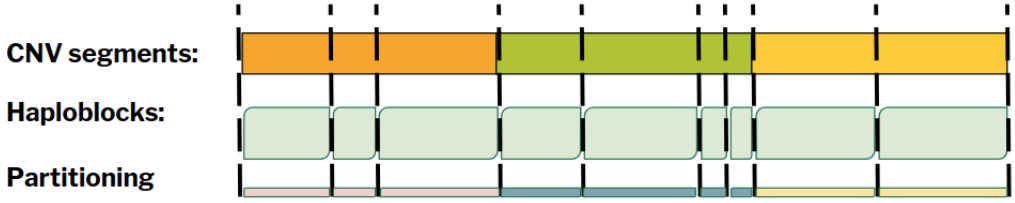


Рис. 2.5: Сегментирование генома в первоначальной версии XClone. Гаплотипические блоки, найденные по данным Oxford Nanopore DNA-seq, пересекаются с диапазонами структурных вариаций, найденных CellRanger DNA.

Тем не менее, такой подход оказался хорош лишь в теории. На практике, найденные таким способом гаплотипические блоки оказывались слишком короткими, порядка нескольких килобаз. Кроме того, распределение их длин имело достаточно большую дисперсию, что доставляло неудобства как при реализации, так и при интерпретации результатов: чем меньше сегмент, тем более зашумленный от него исходит сигнал. Когда таких сегментов много, модель переобучалась на шум в них, что приводило к неадекватному предсказанию меток классов.

2.2.4 Исправление ошибок смены цепи

Поскольку одной из главных задач XClone является предсказание *аллель-специфичных* структурных вариаций в геноме, матрицы AD и DP аллель-специфичных прочтений должны отражать биологию аллельного дисбаланса в клетках образца. Для этого нужно понимать, к какому гаплотипу принадлежит каждый ОНП. В разделе про статистическое гаплотипирование ОНП был сделан акцент на том, что существующие алгоритмы гарантируют только локальную корректность: при использовании алгоритма EAGLE2, следует ожидать, что при разбиении хромосомы на непересекающиеся окна длины 20-50 килобаз все гетерозиготные ОНП

в пределах одного окна будут иметь одинаковый гаплотип, если это на самом деле так. Тем не менее, гаплотипы соседних сегментов с точки зрения алгоритма могут не совпадать даже тогда, когда на самом деле должны. К этому приводят так называемые **ошибки смены цепи** (*switching error*) — спонтанная и неявная замена гаплотипических меток на противоположные внутри алгоритма. Классификацию ошибок смены цепи можно найти в статье [4], цитата из которой приведена ниже:

"Phasing accuracy is typically measured by counting the number of ***'switches'*** *between known maternal and paternal haplotypes that should not occur if individual maternal and paternal chromosomal nucleotide sequence content has been accurately characterized. If an inconsistency is identified, then it is called a 'switch error.' These switch errors manifest themselves as induced and false recombination events in the inferred haplotypes compared with the true haplotypes. To identify* ***switch errors****, the phase of each site is compared with upstream neighboring phased sites. The switch error rate (SER) is defined as the number of switch errors divided by the number of opportunities for switch errors. Switch errors were further classified into three categories:* ***long****,* ***point****, and* ***undetermined****. A long switch appears as a large-scale pseudo recombination event; that is, there are no other switches in the local neighborhood around the long switch (e.g., no other switches within three consecutive heterozygous sites). On the contrary, a small-scale switch error appearing as two neighboring switch errors is considered as a point switch (e.g., two switches within three consecutive heterozygous sites, with the pair of switches counted as a point switch). The remaining switches are considered undetermined (e.g., only two sites phased in a small phasing block, so the switch error could not be classified into long or point)."*

Тем не менее, разбиение генома на фрагменты по 20-50 килобаз непрактично: в силу разреженности данных, это даёт слабый и зашумленный сигнал аллельного дисбаланса. В связи с этим был разработан метод, одновременно решающий обе описанные проблем. На первом шаге алгоритма происходит разбиение генома на непересекающиеся сплошные сегменты длины L. Затем каждые N подряд идущих сегментов объеди-

няются в блок длины NL. В пределах блока переключения моделируются бернуллиевскими случайными величинами, по одной на каждый сегмент. Параметры этих распределений, в свою очередь, выводятся **EM-алгоритмом**. После исправления ошибок, прочтения сегментов внутри блока суммируются, что даёт более стабильный сигнал. Эта идея была сформулирована в [29], но технические детали были осознанно исключены авторами CHISEL из препринта.

Прежде чем приступать к рассмотрению метода, сформулируем необходимые определения:

Определение 2.4 (*EM-алгоритм*)**.**
EM-алгоритм (от английского *"EM"* — *"Expectation Maximization"*) — метод поиска оценок максимального правдоподобия (ОМП) или оценок апостериорного максимума (ОАП) параметров статистических моделей, содержащих скрытые переменные.

Алгоритм 1: EM-алгоритм в общем виде

Результат: $\boldsymbol{\Theta}^*$, $p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\Theta}^*)$
$t = 0$;
$\boldsymbol{\Theta}^{(0)}$ инициализируется случайно;
до тех пор, пока $Q(\boldsymbol{\Theta}^{(t+1)} \mid \boldsymbol{\Theta}^{(t+1)}) - Q(\boldsymbol{\Theta}^{(t)} \mid \boldsymbol{\Theta}^{(t)}) > \varepsilon$
выполнять
  $\mathcal{L}(\boldsymbol{\Theta}^{(t)}; \boldsymbol{Z}, \boldsymbol{X}) := p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\Theta}^{(t)})$;
  $Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)}) := \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\Theta}^{(t)}} \log \mathcal{L}(\boldsymbol{\Theta}; \boldsymbol{Z}, \boldsymbol{X})$ `// E-шаг`
  $\boldsymbol{\Theta}^{(t+1)} := \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)})$ `// M-шаг`
  $t = t + 1$
конец
$\boldsymbol{\Theta}^* := \boldsymbol{\Theta}^{(t)}$

Здесь $\boldsymbol{Z}$ — дискретные скрытые переменные, $\boldsymbol{\Theta}$ — параметры статистической модели, $\boldsymbol{X}$ — выборка, $\varepsilon > 0$, p — функция плотности. Каждая итерация алгоритма состоит из двух основных шагов:

1. **E-шаг**, на котором устраняется явная зависимость от скрытых переменных посредством взятия матожидания логарифма совместной функции правдоподобия по условному распределению $\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\Theta}^{(t)}$;

2. **M-шаг**, на котором параметры нового апостериорного распределения $\boldsymbol{\Theta}^{(t+1)}$ выбираются таким образом, чтобы максимизировать $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)})$ — функцию правдоподобия "в среднем".

С теоретическим обоснованием и формальным доказательством корректности EM-алгоритма можно ознакомиться в ([23], стр. 363-365). В контексте решаемой задачи $\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Theta}$ имеют следующий смысл:

- $\boldsymbol{Z} = \{z_1, \ldots, z_N\}$ — независимые в совокупности индикаторы корректности гаплотипов сегментов

$$\forall i : z_i \sim \text{Bern}(p_i)$$

$$\forall q \in \{0,1\}^N : p(\boldsymbol{Z} = q \mid p_1, \ldots, p_n) = \prod_{i=1}^{N} p(z_i = q_i \mid p_i) = \prod_{i=1}^{N} p_i^{q_i}(1-p_i)^{1-q_i}$$

  Если $z_i = 1$, то будем говорить, что сегмент i имеет корректный гаплотип, иначе — инвертированный. Эти обозначения имеют смысл только в пределах одного блока, в соседних блоках метки могут иметь противоположный смысл. Из этого наблюдения становится ясно, что алгоритм не решает проблему переключения полностью, но уменьшает число ошибок за счёт аггрегации сегментов в блоки.

- Обозначим через M число клеток образца, тогда $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M)$, $X_c := (\boldsymbol{a}_c, \boldsymbol{b}_c)$ — вектора прочтений для каждой из клеток, по компоненте на сегмент. $\boldsymbol{a}_c = (a_{c,1}, \ldots, a_{c,N})$ — число прочтений аллеля А (альтернативный аллель), $\boldsymbol{b}_c = (b_{c,1}, \ldots, b_{c,N})$ — аллеля Б (референсный аллель).

- $\forall c \in \overline{1, M} : \boldsymbol{r}_c := \boldsymbol{a}_c + \boldsymbol{b}_c$ — вектора прочтений обоих аллелей вместе.

- $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_M; p_1, \ldots, p_N)$, где θ_c — пропорция ридов гаплотипа 1 в блоке в клетке c. Алгоритм предполагает, что пропорция гаплотипа 1 одинакова во всех сегментах внутри блока с точностью до переключения.

В этих обозначениях можно сформулировать и доказать следующее утверждение:

Утверждение 2.1. Правила пересчёта параметров апостериорного распределения на M-шаге EM-алгоритма имеют вид:

$$\begin{aligned} p_i^{(t+1)} &= \frac{p_i^{(t)} \prod_{c=1}^{M} (\theta_c^{(t)})^{a_{c,i}} (1-\theta_c^{(t)})^{b_{c,i}}}{p_i^{(t)} \prod_{c=1}^{M} (\theta_c^{(t)})^{a_{c,i}} (1-\theta_c^{(t)})^{b_{c,i}} + (1-p_i^{(t)}) \prod_{c=1}^{M} (\theta_c^{(t)})^{b_{c,i}} (1-\theta_c^{(t)})^{a_{c,i}}} \\ \theta_c^{(t+1)} &= \frac{\sum_{i=1}^{N} a_{i,c}\gamma_{i,1}^{(t)} + b_{i,c}\gamma_{i,0}^{(t)}}{\sum_{i=1}^{N} r_{i,c}} \end{aligned} \tag{2.1}$$

где $\forall j \in \{0,1\} : \gamma_{i,j}^{(t)} := P(z_i = j \mid \boldsymbol{X}, \boldsymbol{\Theta}^{(t)})$.

Доказательство. Вектора прочтений в клетках независимы в совокупности, потому:

$$P(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\Theta}) = \prod_{c=1}^{M} p(\boldsymbol{X}_c \mid \boldsymbol{Z}, \boldsymbol{\Theta}) = \prod_{c=1}^{N} \theta_c^{\widehat{a}_c(\boldsymbol{Z})} (1-\theta_c)^{\widehat{b}_c(\boldsymbol{Z})}$$

Где

$$\begin{cases} \widehat{a}_c(\boldsymbol{Z}) := \sum_{i=1}^{N} \left[z_i a_{c,i} + (1-z_i) b_{c,i} \right], \\ \widehat{b}_c(\boldsymbol{Z}) := \sum_{i=1}^{N} \left[(1-z_i) a_{c,i} + z_i b_{c,i} \right], \\ c \in \overline{1, M} \end{cases}$$

Тогда функция правдоподобия и её логарифм принимают вид

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\Theta}; \boldsymbol{X}, \boldsymbol{Z}) = p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\Theta}) = p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\Theta}) p(\boldsymbol{Z} \mid \boldsymbol{\Theta}) \\ &l(\boldsymbol{\Theta}; \boldsymbol{X}, \boldsymbol{Z}) = \log \mathcal{L}(\boldsymbol{\Theta}; \boldsymbol{X}, \boldsymbol{Z}) = \\ &= \log \prod_{\boldsymbol{q} \in \{0,1\}^N} \left(\prod_{c=1}^{M} \theta_c^{\widehat{a}_c(\boldsymbol{q})} (1-\theta_c)^{\widehat{b}_c(\boldsymbol{q})} \prod_{i=1}^{N} p_i^{q_i} (1-p_i)^{1-q_i} \right)^{\mathbb{I}\{\boldsymbol{Z}=\boldsymbol{q}\}} = \\ &= \sum_{\boldsymbol{q} \in \{0,1\}^N} \mathbb{I}\{\boldsymbol{Z} = \boldsymbol{q}\} \left(\sum_{c=1}^{M} \sum_{i=1}^{N} \widehat{a}_{c,i}(\boldsymbol{q}) \log \theta_c + \widehat{b}_{c,i}(\boldsymbol{q}) \log(1-\theta_c) \right) + \\ &+ \sum_{\boldsymbol{q} \in \{0,1\}^N} \mathbb{I}\{\boldsymbol{Z} = \boldsymbol{q}\} \left(\sum_{i=1}^{N} q_i \log p_i + (1-q_i) \log(1-p_i) \right) \end{aligned}$$

Изменением порядка суммирования можно показать, что каждая из этих двух сумм распадается на N сумм поменьше, по одной на каждую из

скрытых переменных. В следствие этого и того, что компоненты случайного вектора $\boldsymbol{Z}$ независимы в совокупности, шаги EM-алгоритма имеют вид:

E-шаг:

$$p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\Theta}^{(t)}) \propto p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{\Theta}^{(t)}) p(\boldsymbol{Z} \mid \boldsymbol{\Theta}^{(t)}) \implies$$

$$\implies \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\Theta}^{(t)}} l(\boldsymbol{\Theta}; \boldsymbol{Z}, \boldsymbol{X}) = \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{z}_i|\boldsymbol{X}_i,\boldsymbol{\Theta}^{(t)}} \log \mathcal{L}(\boldsymbol{\Theta}; \boldsymbol{z}_i, \boldsymbol{X}_i) =$$

$$= \sum_{i=1}^{N} \sum_{q_i=0}^{1} p(\boldsymbol{z}_i = q_i \mid \boldsymbol{X}_i, \boldsymbol{\Theta}^{(t)}) \left(\begin{array}{l} \sum\limits_{c=1}^{M} \left[\widehat{a}_{c,i}(q_i) \log \theta_c + \widehat{b}_{c,i}(q_i) \log(1-\theta_c) \right] + \\ + \log p(\boldsymbol{z}_i = q_i \mid \boldsymbol{\Theta}) \end{array} \right) =$$

$$= \sum_{i=1}^{N} \left[\begin{array}{l} \gamma_{i,1}^{(t)} \left(\sum\limits_{c=1}^{M} \left[a_{c,i} \log \theta_c + b_{c,i} \log(1-\theta_c) \right] + \log p_i \right) + \\ + \gamma_{i,0}^{(t)} \left(\sum\limits_{c=1}^{M} \left[b_{c,i} \log \theta_c + a_{c,i} \log(1-\theta_c) \right] + \log(1-p_i) \right) \end{array} \right] = Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)})$$

M-шаг:

$$p_i^{(t+1)} = \arg\max_{p_i} Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)}) \iff \frac{\gamma_{i,1}^{(t)}}{p_i^{(t+1)}} - \frac{\gamma_{i,0}^{(t)}}{1 - p_i^{(t+1)}} = 0 \iff p_i^{(t+1)} = \gamma_{i,1}^{(t)}$$

$$\theta_c^{(t+1)} = \arg\max_{\theta_c} Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)}) \iff \frac{\sum_{i=1}^{N} \gamma_{i,1}^{(t)} a_{c,i} + \gamma_{i,0}^{(t)} b_{c,i}}{\theta_c^{(t+1)}} - \frac{\sum_{i=1}^{N} \gamma_{i,1}^{(t)} b_{c,i} + \gamma_{i,0}^{(t)} a_{c,i}}{1 - \theta_c^{(t+1)}} = 0$$

$$\iff \theta_c^{(t+1)} = \frac{\sum_{i=1}^{N} \gamma_{i,1}^{(t)} a_{c,i} + \gamma_{i,0}^{(t)} b_{c,i}}{\sum_{i=1}^{N} a_{c,i} + b_{c,i}}$$

Где необходимое условие локального экстремума является также достаточным в силу выпуклости функции $Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(t)})$ ([23], стр. 363-364). $\square$

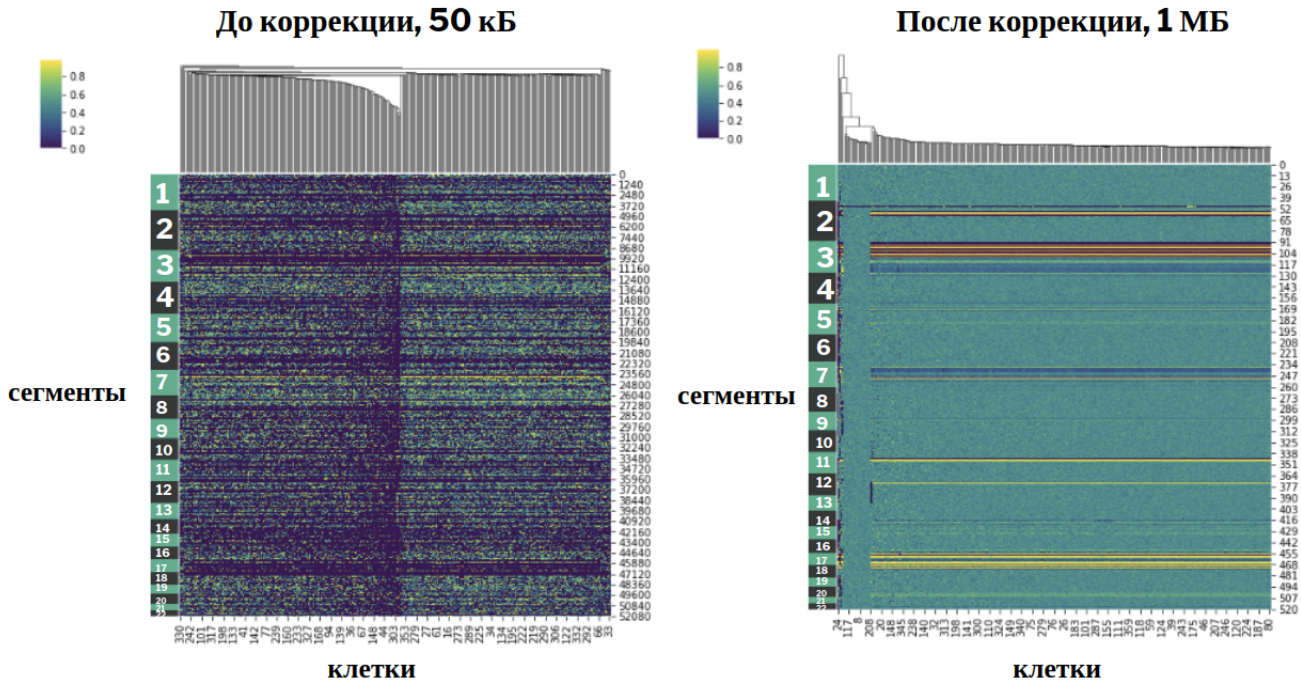


Рис. 2.6: Коррекция ошибок смены цепи на примере ДНК-образца STP-Nuclei. На рисунке изображены тепловые карты долей аллеля из «материнского» гаплотипа, числа слева обозначают номер хромосомы. Картина аллельного дисбаланса до коррекции практически не прослеживается, после — становится очевидна.

После того, как значения $z_1, \ldots, z_N$ были определены по данным DNA-seq, их же можно использовать для предобработки данных RNA-seq, полученных из образцов тканей того же пациента. Это даёт возможность интеграции двух модальностей в единую статистическую модель.

Стоит отметить, что на практике $p_i^{(t+1)}$ следует считать по эквивалетной, но уже численно устойчивой формуле:

$$p_i^{(t+1)} = \left(1 + \exp\left[\log(1 - p_i^{(t)}) - \log(p_i^{(t)}) + \sum_{c=1}^{M} \Delta_{c,i}(\log(\theta_c^{(t)}) - \log(1 - \theta_c^{(t)}))\right]\right)^{-1}$$

Где $\Delta_{c,i} := b_{c,i} - a_{c,i}$, а показатель экспоненты стоит искусственно приводить к диапазону $[-C; C]$ для некоторого $C > 0$ (авторами было выбрано $C = 100$). В противном случае $\prod_{c=1}^{M}(\theta_c^{(t)})^{a_{c,i}}(1 - \theta_c^{(t)})^{b_{c,i}}$ может представлять собой произведение тысяч или даже миллионов очень маленьких

величин в больших степенях. Стандартной реализации чисел с плавающей запятой двойной точности недостаточно для хранения результатов промежуточных вычислений при использовании наивной формулы.

2.3 Использованные данные

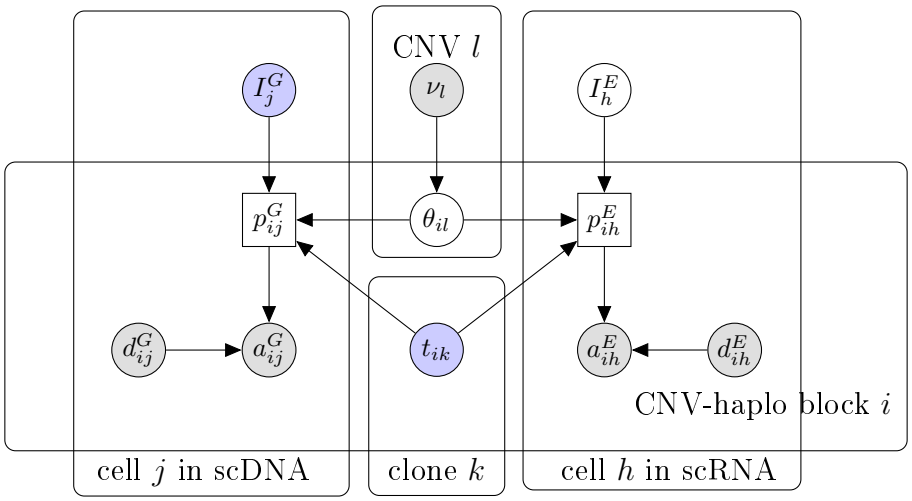
2.4 Первоначальная версия XClone: только ASE-модуль

2.4.1 Plate notation

2.4.2 Семплирование по Гиббсу

2.4.3 Предложенная модель, её недостатки

scDNA + CNV and scRNA, different samples



In this model, clonal assignment $\mathbf{I^G}$ of cells in scDNA sample is assumed fixed. For each clone k, for each same-CNV block i the clonal CNV state $\mathbf{T}_{i,k}$ is defined to be the most frequent CNV state in that position across all cells assigned to clone k. Clonal assignment $\mathbf{I^E}$ in scRNA sample is learned.

This model learns the clonal structure in scRNA sample using ASE profiles from scDNA.

Basic definitions

1. **Constants:**

- M^G, M^E — number of cells in scDNA and scRNA samples respectively.
- K — estimated number of clones in the sample.
- N — number of same-CNV blocks in scDNA sample (also used in scRNA sample).
- $T_{\max}$ — maximal possible CNV number. User-defined with the default of 5.
- τ — set of possible CNV configurations:

$$\{(1,0),(0,1),(2,0),(1,1),(1,2),\ldots,T_{\max},0),\ldots,(0,T_{\max})\}$$

  The case of zero CNV number should be treated with care if we can't say for sure whether the part of the chromosome (both arms) is deleted).

2. **Other known quantities:**

- $\mathbf{D^G}, \mathbf{D^E}$ — total read counts. To get a count of the block, one simply adds up the counts of the variants within the block. Here we assume that variants are far enough from each other, so that almost no reads overlap two variants at the same time. Otherwise, adding things up wouldn't make sense.
- $\mathbf{A^G}$, $\mathbf{A^E}$ — same for allele-specific counts.
- $\mathbf{f} = (f_1, \ldots, f_K)$ — K-dimensional vector of estimated clonal fractions ($\sum_{i=1}^{K} f_i = 1$)
- $\mathbf{T}$ — CNV states of blocks ($\mathbf{T_{i,k}}$ is a CNV state of a block i in clone k).

3. **Inferred quantities:**

- $\mathbf{\Theta}$ — allelic rates of variants located within blocks of fixed CNV status. $\mathbf{\Theta_{i,t}}$ is an allelic rate of a block i with a CNV status t. Set of blocks for each clone is unique. Blocks are ordered as tuples of the form (*block start, block length*).

- $\mathbf{I^E} \in [K]^M$ — cell-to-clone assignment in scRNA sample.

4. **Some notational conventions:**

- Capitalized letter without superscript (like $\mathbf{\Theta}$) denotes the information for both samples.
- $\mathbf{H^G}, \mathbf{H^E}$ — CNV status of the blocks in accordance with the current label assignment:

$$\mathbf{H^G_{i,j}} := \mathbf{T_{i,I^G_j}}, \quad \mathbf{H^E_{i,j}} := \mathbf{T_{i,I^E_j}}$$

- $\mathbf{X^G}$, $\mathbf{X^E}$ — a shortcut to simplify the notation: $\mathbf{X^{G|E}_{i,j}}$ is an allelic rate of a block i in cell j based of the current cell-to-clone label assignment.

$$\mathbf{X^G_{i,j}} := \mathbf{\Theta_{i,H^G_{i,j}}}, \quad \mathbf{X^E_{i,j}} := \mathbf{\Theta_{i,H^E_{i,j}}}$$

Generative model formulation

- **Cell-to-clone assignment posterior:**

$$\mathrm{P}(\mathbf{I^E_j} = k_0 \mid \mathbf{A_j}, \mathbf{D_j}, \mathbf{f}, \mathbf{\Theta}) = \frac{\mathrm{P}(\mathbf{A_j} \mid \mathbf{D_j}, \mathbf{I^E_j} = k_0, \mathbf{\Theta})\mathrm{P}(\mathbf{I^E_j} = k_0 \mid \mathbf{f})}{\sum_{k=1}^{K} \mathrm{P}(\mathbf{A_j} \mid \mathbf{D_j}, \mathbf{I^E_j} = k, \mathbf{\Theta})\mathrm{P}(\mathbf{I^E_j} = k \mid \mathbf{f})} \tag{2.2}$$

- **ASE model:**

$$\begin{aligned} \mathrm{P}(\mathbf{A^G_{i,j}} \mid \mathbf{D^G_{i,j}}, \mathbf{\Theta}) &= \mathrm{Binom}(\mathbf{A^G_{i,j}} \mid \mathbf{D^G_{i,j}}, \mathbf{X^G_{i,j}}) \\ \mathrm{P}(\mathbf{A^E_{i,j}} \mid \mathbf{D^E_{i,j}}, \mathbf{\Theta}) &= \mathrm{Binom}(\mathbf{A^E_{i,j}} \mid \mathbf{D^E_{i,j}}, \mathbf{X^E_{i,j}}) \end{aligned} \tag{2.3}$$

- **ASE likelihood (both terms factorize over variants):**

$$\begin{aligned} \mathrm{P}(\mathbf{A^G_j} \mid \mathbf{D^G_j}, \mathbf{I^G_j} = k^G, \mathbf{\Theta^E}) &= \prod_{i=1}^{N} \mathrm{Binom}(\mathbf{A^G_j} \mid \mathbf{D^G_j}, \mathbf{\Theta_{i,k^G}}) \\ \mathrm{P}(\mathbf{A^E_j} \mid \mathbf{D^E_j}, \mathbf{I^E_j} = k^E, \mathbf{\Theta}) &= \prod_{i=1}^{N} \mathrm{Binom}(\mathbf{A^E_j} \mid \mathbf{D^E_j}, \mathbf{\Theta_{i,k^E}}) \end{aligned} \tag{2.4}$$

- **Allelic rate likelihood:**

$$\mathcal{L}(\mathbf{\Theta}) = \left(\prod_{j=1}^{M^G} \sum_{k^G=1}^{K} \mathrm{P}(\mathbf{A_j^G} \mid \mathbf{D_j^G}, \mathbf{I_j^G} = k^G, \mathbf{\Theta}) \right) \times \\ \times \left(\prod_{j=1}^{M^E} \sum_{k^E=1}^{K} \mathrm{P}(\mathbf{A_j} \mid \mathbf{D_j^E}, \mathbf{I_j^E} = k^E, \mathbf{\Theta}) \cdot \mathrm{P}(\mathbf{I_j^E} = k^E \mid \mathbf{f}) \right) \tag{2.5}$$

To view the clonal assignment in a Bayesian way, we introduce informative prior ν for $\mathbf{\Theta}$.

Using that «posterior $\propto$ prior $\times$ likelihood», we obtain:

$$\mathrm{P}(\mathbf{\Theta} \mid \mathbf{A}, \mathbf{D}, \mathbf{f}, \nu) \propto \mathrm{P}(\mathbf{\Theta} \mid \nu) \times \mathcal{L}(\mathbf{\Theta}) = \\ = \prod_{l=1}^{N} \prod_{t \in \tau} \mathrm{Beta}(\alpha_{l,\tau}, \beta_{l,\tau}) \times \mathcal{L}(\mathbf{\Theta}) \tag{2.6}$$

Parameters α_t, β_t are selected in such a way that the mode of $\mathrm{Beta}(\alpha_t, \beta_t)$ equals to $1/t$ [5].

Selecting a prior for $\mathbf{\Theta}$ CNV state of t hides a plethora of possible configurations: it can mean "k copies of maternal chromosome and $t-k$ copies of paternal"for any $k \in \{0, \ldots, t\}$, all the variants are possible. But they are not equally possible: some are more supported by evidence than the rest. During initialization, for each block in each clone we should find the (k,t)-configuration (k_0, t), such that k_0/t is as close to the observed ASE ratio as possible. Then we choose values (α, β) such that the mode of $\mathrm{Beta}(\alpha, \beta)$, given by $(\alpha-1)/(\alpha+\beta-2)$, equals to k_0/t. That means, we must solve the following problem:

$$\frac{\alpha - 1}{\alpha + \beta - 2} = \frac{k_0}{t},\ \alpha \geqslant 1,\ \beta \geqslant 1$$

Let's derive the solution. If $k_0 = 0$, it is clear that $\alpha = 1$, while any $\beta > 1$

[5] because if we assume that allelic rates only depend on the CNV status t then those rates could be computed as $1/t$

works[6]. Otherwise:

$$\begin{aligned}(\alpha-1)t &= (\alpha+\beta-2)k_0\\ k_0\beta &= (t-k_0)\alpha - t + 2k_0\\ \beta &= \left(\frac{t}{k_0}-1\right)\alpha - \left(\frac{t}{k_0}-2\right)\\ &\implies \alpha = 1+\frac{t-2k_0}{t-k_0},\ \beta = 1\end{aligned} \tag{2.7}$$

As β is linearly dependent from α, any increase in α will pull β up, "sharpening"the shape of the distribution and making it more biased, thereby we decided to choose the minimal feasible α.

Inference (Gibbs sampler) To use a Gibbs sampler, we define conditional probability distribution for each scalar random variable:

Cell-to-clone label assignment:

$$\mathrm{P}(\mathbf{I^E_j} = k \mid \mathbf{I^E_{-j}}, \mathbf{A^E}, \mathbf{D^E}, \mathbf{f}, \boldsymbol{\Theta}) \propto \mathrm{P}(\mathbf{A^E_j} \mid \mathbf{D^E_j}, \mathbf{I^E_j} = k, \boldsymbol{\Theta}) \cdot \mathrm{P}(\mathbf{I^E_j} = k \mid \mathbf{f}) \tag{2.8}$$

Allelic rates: Assuming fixed assignment, let's expand the joint likelihood equation:

$$\begin{aligned}&\mathrm{P}(\boldsymbol{\Theta} \mid \mathbf{A}, \mathbf{D}, \mathbf{I^G}, \mathbf{I^E}, \mathbf{f}, \nu) \propto\\ &\propto \left\{\prod_{l=1}^{N}\prod_{t\in\tau}\mathrm{Beta}(\alpha_{T_{l,t}}, \beta_{T_{l,t}})\right\}\left[\prod_{j^G=1}^{M^G}\mathrm{P}(\mathbf{A^G_{jG}} \mid \mathbf{D^G_{jG}}, \mathbf{I^G_{jG}}, \boldsymbol{\Theta})\right]\left[\prod_{j^E=1}^{M^E}\mathrm{P}(\mathbf{A^E_{jE}} \mid \mathbf{D^E_{jE}}, \mathbf{I^E_{jE}}, \boldsymbol{\Theta})\right] =\\ &= \prod_{l=1}^{N}\prod_{t\in\tau}\mathrm{Beta}(\alpha_{T_{l,t}}, \beta_{T_{l,t}})\left[\prod_{j^G=1}^{M^G}\mathrm{Binom}(\mathbf{A^G_{jG}} \mid \mathbf{D^G_{jG}}, \boldsymbol{\Theta_{i,t}})\right]\left[\prod_{j=1}^{M^E}\mathrm{Binom}(\mathbf{A^E_{jE}} \mid \mathbf{D^E_{jE}}, \boldsymbol{\Theta_{i,t}})\right]\\ &= \prod_{l=1}^{N}\prod_{t\in\tau}\left[\mathrm{Beta}(\alpha_{T_{l,t}}, \beta_{T_{l,t}})\prod_{j^G=1}^{M^G}\prod_{j^E=1}^{M^E}\left(\begin{array}{l}\mathrm{Binom}(\mathbf{A^G_{l,jG}} \mid \mathbf{D^G_{l,jG}}, \mathbf{X^G_{l,jG}})^{\mathbb{I}\left\{\mathbf{H^G_{l,jG}}=t\right\}}\times\\ \times\ \mathrm{Binom}(\mathbf{A^E_{l,jE}} \mid \mathbf{D^E_{l,jE}}, \mathbf{X^E_{l,jE}})^{\mathbb{I}\left\{\mathbf{H^E_{l,jE}}=t\right\}}\end{array}\right)\right]\end{aligned} \tag{2.9}$$

From here we derive update rules for individual allelic rates:

$$\boldsymbol{\Theta_{l,t}} \mid \mathbf{I^G}, \mathbf{I^E} \sim \mathrm{Beta}(\alpha_{T_{l,t}} + u_{l,t}, \beta_{T_{l,t}} + v_{l,t}) \tag{2.10}$$

[6]Nevertheless, it is not clear which one to choose. As we try to reduce prior bias, let's set it to be equal $1+\varepsilon$ for some reasonable $\varepsilon > 0$

where

$$u_{l,t} = \sum_{j^G=1}^{M} \mathbf{A}^{\mathbf{G}}_{\mathbf{l,j^G}} \cdot \mathbb{I}\left\{\mathbf{H}^{\mathbf{G}}_{\mathbf{l,j^G}} = t\right\} + \sum_{j^E=1}^{M} \mathbf{A}^{\mathbf{E}}_{\mathbf{l,j^E}} \cdot \mathbb{I}\left\{\mathbf{H}^{\mathbf{E}}_{\mathbf{l,j^E}} = t\right\}$$
$$v_{l,t} = \sum_{j^G=1}^{M^G} (\mathbf{D}^{\mathbf{G}}_{\mathbf{l,j^G}} - \mathbf{A}^{\mathbf{G}}_{\mathbf{l,j^G}}) \cdot \mathbb{I}\left\{\mathbf{H}^{\mathbf{G}}_{\mathbf{l,j^G}} = t\right\} + \sum_{j^E=1}^{M} (\mathbf{D}^{\mathbf{E}}_{\mathbf{l,j^E}} - \mathbf{A}^{\mathbf{E}}_{\mathbf{l,j^E}}) \cdot \mathbb{I}\left\{\mathbf{H}^{\mathbf{E}}_{\mathbf{l,j^E}} = t\right\} \tag{2.11}$$

2.4.4 Поиск наиболее вероятной перестановки меток

При валидации модели на синтетических данных ключевым предсказываемым объектом было распределение вероятностей на K клональных метках — матрица

$$\boldsymbol{P} \in \mathbb{R}_{+}^{M \times K}, \forall i \in [M] : \sum_{j=1}^{K} p_{i,j} = 1$$

Тем не менее, модель предсказывала метки с точностью до неизвестной перестановки. Если предсказание точное, т.е. что $\boldsymbol{P}$ с точностью до перестановки совпадает с истинной, которая задаётся бинарной матрицей $\boldsymbol{Q}$, то сама перестановка определяется легко: биекция между столбцами $\boldsymbol{P}$ и $\boldsymbol{Q}$ тривиально строится за $O(K(K+M))$. Но на практике модель ошибается: либо не может восстановить метку, либо не успевает это сделать за отведённое число итераций. Определения качества предсказания в таком случае — нетривиальная задача. Перебор всех возможных перестановок столбцов, коих $K!$, и выбор той, на которой достигается минимальное расстояние между столбцами матриц $\boldsymbol{P}$ и $\boldsymbol{Q}$, реализуем при малых K. Тем не менее, сверхполиномиальная асимптотика не позволяет масштабировать алгоритм: проверить работоспособность модели было бы невозможно даже при $M = 10^4$ и $K = 10$ наивный алгоритм потребовал бы порядка $O(K!M)$ операций, т.е. порядка 4×10^{10}. При этом в открытом доступе опубликованы образцы с $M = 1.3 \times 10^{67}$, и типичное

[7]https://www.10xgenomics.com/blog/our-13-million-single-cell-dataset-is-ready-to-download

количество клеток в данных от 10X Genomics от года к году монотонно увеличивается.

Для решения этой проблемы был разработан полиномиальный алгоритм, находящий оптимальную перестановку за $O(K^4)$ операций и $O(K^2)$ дополнительной памяти. Алгоритм основан на сведении к задаче поиска в двудольном графе совершенного паросочетания минимального веса. А именно: строится взвешенный полный двудольный граф $K_{K\times K}$, столбцам $\boldsymbol{P}$ сопоставляются вершины левой доли, $v_1, \ldots, v_K$, столбцам $\boldsymbol{Q}$ — вершины правой доли, $u_1, \ldots, u_K$, а ребру (v_i, u_j) — вес, равный $d(P_i, Q_j)$, где P_i, Q_j — соотв. столбцы, а d — метрика (значение по умолчанию — l_1-норма). То, что совершенному паросочетанию в таком графе соответствует именно $\arg\min_{\sigma\in S_K} \sum_{j=1}^{K} d(P_j, Q_j)$, очевидно по построению. Задача поиска совершенного паросочетания минимального веса в двудольном графе решается — это т.н. **задача о назначениях**, одна из фундаментальных задач кобминаторной оптимизации. Для её решения применяется так называем "венгерский алгоритм опубликованный в 1955 году американским математиком Гарольдом Куном[12].

Рис. 2.7: Иллюстрация сведения задачи восстановления наиболее вероятной перестановки меток классов к поиску совершенного паросочетания минимального веса в двудольном графе. Веса — расстояния между столбцами матриц $\boldsymbol{P}$ и $\boldsymbol{Q}$, матриц предсказанных и истинных вероятностей клональных меток.

2.5 Заключительная версия XClone: BAF- и RDR-модули

Текущая версия алгоритма XClone использует BAF и RDR для того, чтобы восстановить клональную структуру опухоли и определить профили ASCNV для каждого клона. Она лишена сразу нескольких недостатков прошлой версии:

- не зависит от внешних алгоритмов детектирования CNV по данным single-cell секвенирования;

- для поиска оценки апостериорного максимума вместо медленного семплирования по Гиббсу используется эффективно автоматизированный вариационный байесовский вывод на GPU;

От пользователя требуется только предоставить BAM-файлы образцов и набор геномных позиций, использованных при секвенировании. Вся предобработка данных и дальнейшее обучение модели релизовано в виде отдельных скриптов с консольным интерфейсом, для использования которых достаточно задать пути к данным.

Сам алгоритм принимает на вход три матрицы:

- $\boldsymbol{RD}$ — матрица прочтений;
- $\boldsymbol{DP}$ — матрица прочтений, которые выравниваются на последовательности, содержащие хотя бы один ОНП;
- $\boldsymbol{AD}$ — матрица тех прочтений из $\boldsymbol{DP}$, которые выравниваются на материнский аллель;

Все три матрицы принадлежат $\mathbb{N}^{N \times M}$, где N это число блоков после предобработки из раздела 2.2.4, а M — число клеток образца. Все три матрицы должны быть подсчитаны по данным scDNA-seq.

Также на вход алгоритма подаётся K — ожидаемое число клональных линий в образце — и τ, набор допустимых ASCNV $\{c_t\}_{t=1}^{T}$, где $c_t := (c_{t,m}, c_{t,p})$, где $c_{t,m}$ — число копий на материнской хромосоме, а $_{t,p}$ — соответственно, на отцовской. По умолчанию для τ содержит следующие 16 состояний:

$$
\begin{aligned}
&([0,0], [0,1], [0,2], [0,3],\\
&[1,0], [1,1], [1,2], [1,3],\\
&[2,0], [2,1], [2,2], [2,3],\\
&[3,0], [3,1], [3,2], [3,3])
\end{aligned}
$$

BAF- и RDR-модули связаны скрытыми переменными $\boldsymbol{Z}, \boldsymbol{Y}$:

- $z_{j,k} := I\{\text{клетка } j \text{ принадлежит клональной линии } k\}$;

- $y_{i,k,t} := I\{\text{блок } i \text{ в клональной линии } k \text{ находится в состоянии } c_t\}$;

на которых заданы априорные вероятности $\boldsymbol{\pi}, \boldsymbol{U}$:

$$\begin{aligned} p(z_{j,k} = 1 \mid \boldsymbol{\pi}) &= \texttt{Multinom}(1; \boldsymbol{\pi}_j) = \pi_{j,k} \\ p(y_{i,k,t} = 1 \mid \boldsymbol{U}) &= \texttt{Multinom}(1; \boldsymbol{u}_{i,k}) = u_{i,k,t} \end{aligned} \tag{2.12}$$

2.5.1 Структура BAF-модуля

Если блок i в клетке j находится в состоянии c_t, то $a_{i,j}$ подчиняется биномиальной модели с параметром $\theta_{i,t} \sim \text{Beta}(\alpha_t, \beta_t)$, где α_t, β_t — параметры априорного распределения. Они одни и те же для всех блоков в состоянии c_t и полагаются равными $\alpha_t = (c_{t,1} + 0.01)$ и $\beta_t = (c_{t,2} + 0.01)$. В формульной записи:

$$p(a_{i,j} \mid d_{i,j}, \theta_{i,t}) = \texttt{Binom}(a_{i,j} \mid d_{i,j}, \theta_{i,t}). \tag{2.13}$$

где $a_{i,j}, d_{i,j}$ — элементы матриц $\boldsymbol{AD}, \boldsymbol{DP}$ на позиции (i, j). В предположении, что числа прочтений в соседних блоках независимы, функция правдоподобия записывается следующим образом:

$$p(A, D|Z, Y, \boldsymbol{\theta}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \prod_{k=1}^{K} \prod_{t=1}^{T} p(a_{i,j} \mid d_{i,j}, \theta_{i,t})^{z_{j,k} \times y_{i,k,t}} \tag{2.14}$$

2.5.2 Структура RDR-модуля

Для каждой клональной линии k модель числа прочтений задаётся мультиномиальным распределением. Каждому блоку i в этой модели назначается вероятность $f_{i,k}$. Она задаётся формулой

$$f_{i,k} := \frac{\sum_{t=1}^{T} m_i \exp[\gamma_t] P(y_{i,k,t} = 1)}{\sum_{b=1}^{N} \sum_{t=1}^{T} m_b \exp[\gamma_t] P(y_{b,k,t} = 1)} \tag{2.15}$$

Где m_i — доля прочтений, попадающих в блок i в здоровой клетке при стремлении глубины покрытия к бесконечности (или какая-то оценка этой величины), $\exp\{\gamma_t\}$ — амплификация, вызванная наличием $c_{t,m} + c_{t,p}$ копий блока в геноме (здесь возведение в степень нужно для того, чтобы гарантировать неотрицательность). Эта модель согласуется с теорией

для данных scDNA-seq, т.к. после коррекции технических факторов для ДНК-секвенирования правомерны предположения о том, что прочтения распределяются по геному равномерно, а систематические отклонения от этого правила могут быть вызваны только loss- или gain-событиями (см. модель ДНК-секвенирования из раздела 2.1).

Тогда функция правдоподобия запишется как

$$\begin{aligned} \boldsymbol{r}_j &:= (r_{1,j}, \dots, r_{N,j}) \\ p(\boldsymbol{r}_j \mid \boldsymbol{f}_k) &:= \texttt{Multinom}\left(\boldsymbol{r}_j \;\middle|\; \sum_{i=1}^{N} r_{i,j}, \boldsymbol{f}_k\right) \\ p(\boldsymbol{RD} \mid \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{m}, \boldsymbol{\gamma}) &:= \prod_{j=1}^{M}\prod_{k=1}^{K} p\left(\boldsymbol{r}_j \;\middle|\; \sum_{i=1}^{N} r_{i,j}, \boldsymbol{f}_k\right)^{z_{j,k}} \end{aligned} \tag{2.16}$$

Если $\boldsymbol{m}$ заранее неизвестно, то его можно смоделировать посредством задания априорного распределения $\texttt{Dirichlet}(\boldsymbol{\omega})$. Аналогично для γ в качестве априорного распределения задаётся $\texttt{GP}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, где под GP подразумевается дискретный гауссовский процесс с $\mu_t := \log((c_{t,m} + c_{t,p})/2 + \varepsilon)$ для достаточного малого положительного ε, а матрица ковариаций $\boldsymbol{\Sigma}$ задаётся RBF-ядром $K(\boldsymbol{c}_t, \boldsymbol{c}_{t\prime}) := l_1 \exp\{-l_2[(c_{t,m} - c_{t\prime,m})^2 + (c_{t,p} - c_{t\prime,p})^2]\}$, где l_1, l_2 — гиперпараметры модели. В формульной записи:

$$\begin{aligned} \boldsymbol{m} \mid \boldsymbol{\omega} &\sim \texttt{Dirichlet}(\boldsymbol{\omega}) \\ \boldsymbol{\gamma} \mid \boldsymbol{\mu}, \Sigma &\sim \texttt{GP}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{aligned} \tag{2.17}$$

2.5.3 Вариационный байесовский вывод

2.5.3.1 Общее описание метода

Пусть $\boldsymbol{X} = (X_1, \dots, X_n)$ — выборка из распределения с плотностью $p_\theta(\boldsymbol{X})$ ($p_\theta(\boldsymbol{X}) = \prod_{i=1}^{n} p_\theta(X_i)$), где θ — набор параметров распределения. Пусть q — априорное распределение на Θ — множестве допустимых параметров. Тогда $f(\theta, \boldsymbol{X}) := p_\theta(\boldsymbol{X})q(\theta)$ — плотность совместного распределения на $\Theta \times \mathcal{X}$. Тогда апостериорная плотность на параметрах выражается как

$$q(\theta \mid \boldsymbol{X}) := \frac{f(\theta, \boldsymbol{X})}{p(\boldsymbol{X})} = \frac{f(\theta, \boldsymbol{X})}{\int_\Theta f(t, \boldsymbol{X})dt} = \frac{f(\theta, \boldsymbol{X})}{\int_\Theta p_t(\boldsymbol{X})q(t)dt}$$

Величину $p(\boldsymbol{X})$ в знаменателе называют **обоснованностью**. В общем случае она не выражается в аналитических функциях.

Пусть стоит задача найти оценку апостериорного максимума (ОАМ) — $\arg\max_{\Theta} q(\theta \mid \boldsymbol{X})$, — а также распределение на всех параметрах из θ. В силу того, обоснованность $p(\boldsymbol{X})$ это константа, можно максимизировать не $q(\theta \| \boldsymbol{X})$, а $f(\theta, \boldsymbol{X})$. Но совместная плотность как функция может быть устроена произвольно сложно. Уже при счётном Θ поиск её максимума это нетривиальная задача. Тем не менее, задача поиска ОАМ допускает альтернативную формулировку в форме задачи непрерывной оптимизации.

Определение 2.5 (*KL-дивергенция*)**.** Пусть p, q — плотности распределений P, Q над одним вероятностным пространством $(\Omega, \mathcal{F}, \mathrm{P})$. Тогда KL-дивергенцией $KL(P \parallel Q)$ называют величину $\int_\Omega p(x) \log \frac{p(x)}{q(x)} dx$.

Утверждение 2.2. Пусть $\mathbb{F}_{\mathcal{X}}$ — пространство абсолютно непрерывных распределений над вероятностным пространством $\mathcal{X} = (\Omega, \mathcal{F}, \mathrm{P})$, тогда

1. $\forall P, Q \in \mathbb{F}_{\mathcal{X}} : KL(P \parallel Q) \geqslant 0$, причём $KL(P \parallel Q) = 0 \iff P \overset{\text{п.вс.}}{=} Q$.
2. $\forall P_1, P2, Q \in \mathbb{F}_{\mathcal{X}} : KL(P_1 \parallel P_2) \leqslant KL(P_1 \parallel Q) + KL(Q \parallel P_2)$

Кроме того, для произвольного $\mathcal{X}$ могут найтись $P, Q \in \mathbb{F}_{\mathcal{X}}$, такие что $KL(P \parallel Q) \neq KL(Q \parallel P)$, что не позволяет считать KL-дивергенцию метрикой на на $\mathbb{F}_{\mathcal{X}}$.

Вариационный байесовский вывод (далее — VB) основан на следующей идее: заменить апостериорное распределение $Q(\theta \mid \boldsymbol{X})$, которое может быть устроено сколь угодно сложно, на удобную для оптимизации аппроксимацию $Q^*(\theta)$ из семейства распределений $\mathcal{Q}$. Тогда есть следующий алгоритм приближённого поиска ОМА:

Алгоритм 2: Вариационный байесовский вывод, общий вид

Результат: θ_0^*, Q_0^* — приближение ОМА и соотв. аппроксимация

$Q_0^* := \arg\min_{Q^* \in \mathcal{Q}} KL(Q^* \parallel Q(\cdot \mid \boldsymbol{X}))$;

$\theta_0^* := \arg\max_{\theta \in \Theta} Q_0^*(\theta)$

Этот метод центральный для алгоритма XClone, потому в этом разделе приведен его подробный обзор.

Определение 2.6 (*ELBO*)**.** В обозначениях данного раздела, величину

$$\mathcal{L}(Q^*) := \int_\Theta q^*(\theta) \log \frac{f(\theta, \boldsymbol{X})}{q^*(\theta)} d\theta$$

называют ELBO — evidence lower bound. На русский язык это можно перевести как «нижняя оценка логарифма обоснованности», но устоявшегося перевода в сообществе нет, потому обычно используют сокращение ELBO.

Утверждение 2.3. Максимизация ELBO по $Q^* \in \mathcal{Q}$ эквивалентна минимизации $KL(Q^* \parallel Q_{\boldsymbol{X}})$, где $Q_{\boldsymbol{X}}(\theta) Q(\cdot \mid \boldsymbol{X})$.

Доказательство. Распишем $KL(Q^* \parallel Q_{\boldsymbol{X}})$:

$$\begin{aligned} KL(Q^* \parallel Q_{\boldsymbol{X}}) &= \mathbb{E}_{\varphi \sim Q^*} \log \frac{q^*(\varphi)}{q_{\boldsymbol{X}}(\varphi)} = \\ &= -\mathbb{E}_{\varphi \sim Q^*} \log \frac{q_{\boldsymbol{X}}(\varphi)}{q^*(\varphi)} = \\ &= -\mathbb{E}_{\varphi \sim Q^*} \log \frac{f(\varphi, \boldsymbol{X})}{p(\boldsymbol{X}) q^*(\varphi)} = \\ &= -\mathbb{E}_{\varphi \sim Q^*} \log \frac{f(\varphi, \boldsymbol{X})}{q^*(\varphi)} + \mathbb{E}_{\varphi \sim Q^*} \log p(\boldsymbol{X}) = \\ &= -\mathcal{L}(Q^*) + \log p(\boldsymbol{X}) \end{aligned} \tag{2.18}$$

Отсюда получаем, что $\log p(\boldsymbol{X}) = L(Q^*) + KL(Q^* \parallel Q_{\boldsymbol{X}})$. Логарифм обоснованности не зависит от θ, потому максимизация $\mathcal{L}(Q^*)$ эквивалентна минимизации $KL(Q^* \parallel Q_{\boldsymbol{X}})$, а потому и решению задачи вариационного байесовского вывода.

ELBO можно получить и иначе, расписав $\log p(\boldsymbol{X})$:

$$\begin{aligned}\log p(\boldsymbol{X}) &= \log \int_{\Theta} f(\theta, \boldsymbol{X}) d\theta \\ &= \log \int_{\Theta} f(\theta, \boldsymbol{X}) \frac{q^*(\theta)}{q^*(\theta)} d\theta = \\ &= \log \mathbb{E}_{\varphi \sim Q^*} \frac{f(\varphi, \boldsymbol{X})}{q^*(\varphi)} = \\ &= \Big[\text{Неравенство Йенсена}\Big] \geqslant \\ &\geqslant \mathbb{E}_{\varphi \sim Q^*} \log \frac{f(\varphi, \boldsymbol{X})}{q^*(\varphi)} = \mathcal{L}(Q^*)\end{aligned} \tag{2.19}$$

Такое доказательство не проясняет связь с VB, но полезно в контексте других приложений и помогает понять, почему эта величина имеет именно такое название. □

Замечание 2.1. В силу того, что KL-дивергенция не симметрична, вид оптимальной аппроксимации может противоречить интуитивному представлению о том, как она должна быть устроена. Для лучшего понимания полезно рассмотреть несколько примеров из [23] (стр. 734), которые иллюстрируют разницу между $KL(Q^* \parallel Q_{\boldsymbol{X}})$ — *forward KL* — и $KL(Q_{\boldsymbol{X}} \parallel Q^*)$ — *reverse KL*.

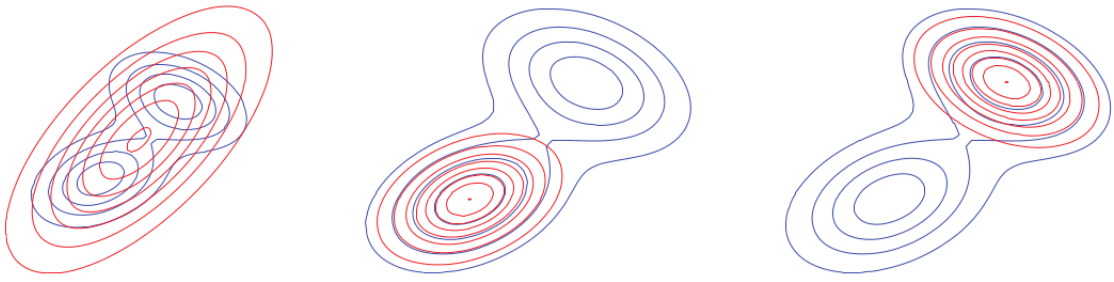


Рис. 2.8: Аппроксимация бимодального распределения Q (синие линии) двумерным гауссовским распределением Q^* (красные линии). (a) — результат минимизации $KL(Q \parallel Q^*)$, (b)-(c) — варианты оптимального приближения при минимизации $KL(Q^* \parallel Q)$

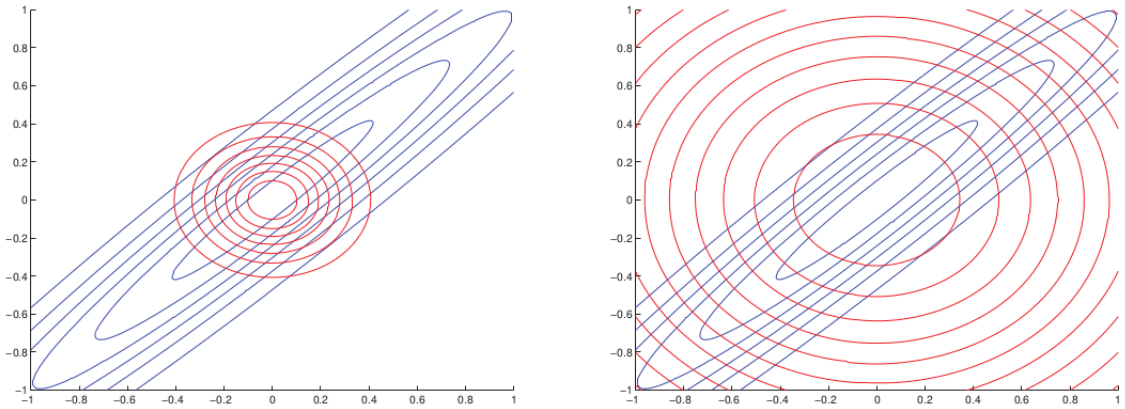


Рис. 2.9: Аппроксимация распределения Q симметричным двумерным гауссовским распределением Q^*. (a) — результат минимизации $KL(Q^* \parallel Q)$, (b) — $KL(Q \parallel Q^*)$

Видно, что при минимизации reverse-KL локальные максимумы аппроксимации Q^* скорее всего будут совпадать с таковыми у Q. При этом может быть так, что Q^* будет хорошо приближать Q только на каком-то множестве, которому распределение Q сопоставляет большую вероятность, потому её называют **zero-forcing**. Минимизация же forward KL будет стремиться хорошо объяснить всё распределение Q в целом, а не только отдельные, важные фрагменты, её называют **zero-avoiding**. В обоих случаях наблюдаемое поведение связано с тем, какая плотность оказывается в знаменателе отношения под логарифмом.

Указанная в предыдущем замечании особенность **очень важная**. Она показывает, что VB, в отличие от MCMC, может даже в пределе не давать возможности семплировать из истинного распределения. Тем не менее, VB существенно проще вычислительно, так как это оптимизационная задача, для решения которой есть хорошо разработанные методы стохастической и распределённой оптимизации. Более того, часто одного только VB бывает достаточно. Вопрос о том, когда использовать VB, а когда MCMC, хорошо резюмирует приведенная цитата из [2]:

«Thus, variational inference is suited to large data sets and scenarios where we want to quickly explore many models; MCMC is suited to smaller

data sets and scenarios where we happily pay a heavier computational cost for more precise samples. For example, we might use MCMC in a setting where we spent 20 years collecting a small but expensive data set, where we are confident that our model is appropriate, and where we require precise inferences. We might use variational inference when fitting a probabilistic model of text to one billion text documents and where the inferences will be used to serve search results to a large population of users. In this scenario, we can use distributed computation and stochastic optimization to scale and speed up inference, and we can easily explore many different models of the data.»

Замечание 2.2. ELBO можно переписать как $\mathcal{L}(Q^*) = \mathbb{E}_{\varphi \sim Q^*} f(\varphi, \boldsymbol{X}) - H(Q^*)$, где H — энтропия. Если $Q^* = Q_{\boldsymbol{X}}$, то левое слагаемое это в точности величина, которая максимизируется на M-шаге EM-алгоритма. Это сходство неслучайно и поясняется в [2]: *«Unlike variational inference, EM assumes the expectation over posterior distribution of latent variables is computable and uses it in otherwise difficult parameter estimation problems. Unlike EM, variational inference does not estimate fixed model parameters — it is often used in a Bayesian setting where classical parameters are treated as latent variables. Variational inference applies to models where we cannot compute the exact conditional of the latent variables.»*

Чаще всего $\mathcal{Q}$ выбирают таким, чтобы распределения из него факторизуются по параметрам, т.е. что

$$q^*(\theta) = \prod_{i=1}^{m} q_i^*(\theta_i)$$

где m это число параметров. Это т.н. **mean field approximation** — метод, вдохновлённый т.н. моделью Изинга из статистической физики. В [2] (стр. 9-10) описан эффективный алгоритм оптимизации маржинальных плотностей q_i^*. Тем не менее, в модели XClone он не используется, т.к. максимизация ELBO происходит автоматически за счёт использования примитивов из Tensorflow.Distributions[7]. При этом $\mathcal{L}(Q^*)$ оптимизируется неявно, посредством максимизации правой части равенства

$\mathcal{L}(Q^*) = KL(Q^* \parallel Q) + \log p(\boldsymbol{X})$, т.к. использование сопряжённых распределений позволяет расписать её в пригодном для покоординатной оптимизации виде.

2.5.3.2 VB в алгоритме XClone

Апостериорные распределения на параметрах модели XClone имеют вид

$$\begin{aligned} & p(\boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\gamma} \mid \boldsymbol{AD}, \boldsymbol{DP}, \boldsymbol{RD}) \propto \\ & \propto p(\boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\gamma}) p(\boldsymbol{AD}, \boldsymbol{DP}, \boldsymbol{RD} | \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\gamma}) = \\ & = p(\boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\gamma}) \cdot \underline{p(\boldsymbol{AD}, \boldsymbol{DP} | \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta})} \cdot \underline{p(\boldsymbol{RD} \mid \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{m}, \boldsymbol{\gamma})} \end{aligned} \tag{2.20}$$

где функция правдоподобия распадается в произведение по BAF- и RDR-модулям, т.к. матрицу $\boldsymbol{RD}$ можно по смыслу считать независищяей от матриц $\boldsymbol{AD}, \boldsymbol{DP}$ по смыслу. Строго говоря, между $\boldsymbol{DP}$ и $\boldsymbol{RD}$ есть положительная корреляция, но она не добавляет новой информации в модель, потому ею было решено пренебречь. Скрытые переменные из $\Omega := \{\boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{m}, \boldsymbol{\gamma}\}$ независимы в совокупности. Апостериорное распределенияе $p(\Omega \mid \boldsymbol{AD}, \boldsymbol{DP}, \boldsymbol{RD})$, как следствие, факторизуется по скрытым переменным. В силу того, что в модели всюду используются сопряжённые распределения, апостериорное распределение имеет тот же вид, что и априорное. Тем не менее, явно вычислить его параметры затруднительно, проще выразить VB через примитивы Tensorflow.Distributions[7] и найти ОАМ стохастическим градиентным спуском в пространстве параметров.

Стоит отметить, что ELBO в стат.модели XClone по явной формуле (через матожидание) оптимизировать сложнее, чем его представление через разность логарифма обоснованности и KL-дивергенции.

$$\mathtt{L}(q) = \int q(\Omega) \log p(\boldsymbol{AD}, \boldsymbol{DP}, \boldsymbol{RD} \mid \Omega) \mathrm{d}\Omega - \mathcal{KL}(q(\Omega) \parallel p(\Omega)) \tag{2.21}$$

Логарифм обоснованности распадается в сумму по BAF- и RDR-модулям.

$$\begin{aligned}&\int q(\Omega)\log p(\boldsymbol{AD},\boldsymbol{DP},\boldsymbol{RD}\mid\Omega)\mathrm{d}\Omega=\\&=\int q(\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta})\log p(\boldsymbol{AD},\boldsymbol{DP}\mid\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta})\mathrm{d}\boldsymbol{Z}\mathrm{d}\boldsymbol{Y}\mathrm{d}\boldsymbol{\theta}+\\&+\int q(\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma})\log p(\boldsymbol{RD}\mid\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma})\mathrm{d}\boldsymbol{m}\mathrm{d}\boldsymbol{\gamma}\end{aligned}\tag{2.22}$$

2.5.3.3 Вывод ELBO для BAF-модуля

Утверждение 2.4. Лог-боснованность BAF-модуля можно расписать как:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta}}\log p(\boldsymbol{AD},\boldsymbol{DP}\mid\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta})=\\&=\int q(\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta})\log p(\boldsymbol{AD},\boldsymbol{DP}\mid\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta})\mathrm{d}\boldsymbol{Z}\mathrm{d}\boldsymbol{Y}\mathrm{d}\boldsymbol{\theta}\\&=\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\sum_{t=1}^{T}\left\{\tilde{\pi}_{j,k}\tilde{u}_{i,k,t}[w_{i,j}+a_{i,j}\psi(\tilde{\alpha}_t)+b_{i,j}\psi(\tilde{\beta}_t)-d_{i,j}\psi(\tilde{\alpha}_t+\tilde{\beta}_t)]\right\}\end{aligned}\tag{2.23}$$

где $b_{i,j}:=d_{i,j}-a_{i,j}$, $\tilde{\cdot}$ указывает, что имеется в виду аппроксимация истинной апостериорной вероятности, $w_{i,j}:=\log\binom{d_{i,j}}{a_{i,j}}$, $\psi(\cdot)$ — дигамма-функция, $\psi(z):=\frac{\Gamma'(z)}{\Gamma(z)}=-\gamma+\int_0^1\left(\frac{1-t^z}{1-t}\right)dt$, где Γ это гамма-функция Эйлера, а γ это константа Эйлера-Маскерони, $\gamma:=\lim\limits_{n\to\infty}\left(-\ln n+\sum_{k=1}^{n}\frac{1}{k}\right)$.

Доказательство. BAF-модуль эквивалентен модели [28] и имеет анало-

гичное доказательство.

$$
\begin{aligned}
&\int q(\boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}) \log p(\boldsymbol{AD}, \boldsymbol{DP} \mid \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}) \mathrm{d}\boldsymbol{Z} \mathrm{d}\boldsymbol{Y} \mathrm{d}\boldsymbol{\theta} \\
&= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta}} \left[\log p(\boldsymbol{AD}, \boldsymbol{DP} \mid \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{\theta}) \right] \\
&= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta}} \left[\log \prod_{i=1}^{N} \prod_{j=1}^{M} \prod_{k=1}^{K} \prod_{t=1}^{T} p(a_{i,j} \mid d_{i,j}, \theta_t)^{z_{j,k} \times y_{i,k,t}} \right] \\
&= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta}} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{T} [z_{j,k} y_{i,k,t} \log \mathtt{Binom}(a_{i,j} \mid d_{i,j}, \theta_t)] \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{\theta}} [z_{j,k} y_{i,k,t} \log \mathtt{Binom}(a_{i,j} \mid d_{i,j}, \theta_t)] \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{Z}}[z_{j,k}] \mathbb{E}_{\boldsymbol{Y}}[y_{i,k,t}] \mathbb{E}_{\boldsymbol{\theta}} [\log \mathtt{Binom}(a_{i,j} \mid d_{i,j}, \theta_t)] \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{T} \left\{ \tilde{\pi}_{j,k} \tilde{u}_{i,k,t} \mathbb{E}_{\boldsymbol{\theta}} [\log \mathtt{Binom}(a_{i,j} \mid d_{i,j}, \theta_t)] \right\} \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{T} \left\{ \tilde{\pi}_{j,k} \tilde{u}_{i,k,t} [w_{i,j} + a_{i,j} \psi(\tilde{\alpha}_t) + b_{i,j} \psi(\tilde{\beta}_t) - d_{i,j} \psi(\tilde{\alpha}_t + \tilde{\beta}_t)] \right\}
\end{aligned}
$$

Где матожидание логарифма плотности бета-биномиального распределения, имеет следующие представление, задействующее дигамма-функцию:

$$
\begin{aligned}
&\mathbb{E}_{\alpha,\beta}[\log q(a \mid d, \theta)] = \\
&= \mathbb{E}_{\alpha,\beta}[\log \mathtt{Binom}(a; d, \theta)] = \\
&= \mathbb{E}_{\alpha,\beta} \left[\log \binom{a}{d} + a \log \theta + (d - a) \log (1 - \theta) \right] = \\
&= \log \binom{a}{d} + a \mathbb{E}_{\theta}[\log \theta] + (d - a) \mathbb{E}_{\beta,\alpha}[\log (1 - \theta)] = \\
&= \log \binom{a}{d} + a(\psi(\alpha) - \psi(\alpha + \beta)) + (d - a)(\psi(\beta) - \psi(\alpha + \beta)) = \\
&= \log \binom{a}{d} + a\psi(\alpha) + (d - a)\psi(\beta) - d\psi(\alpha + \beta)
\end{aligned}
$$

Где тождество $\mathbb{E}_{\alpha,\beta}[\log\theta] = \psi(\alpha) - \psi(\alpha+\beta)$ доказывается так:

$$\begin{aligned}
\mathbb{E}_{\alpha,\beta}\log\theta = &= \int_0^1 \ln x\, \mathtt{Beta}(x;\alpha,\beta)dx = \\
&= \int_0^1 \ln x\, \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathtt{B}(\alpha,\beta)}dx = \\
&= \frac{1}{\mathtt{B}(\alpha,\beta)} \int_0^1 \frac{\partial x^{\alpha-1}(1-x)^{\beta-1}}{\partial\alpha}dx = \\
&= \frac{1}{\mathtt{B}(\alpha,\beta)} \frac{\partial}{\partial\alpha} \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx = \\
&= \frac{1}{\mathtt{B}(\alpha,\beta)} \frac{\partial \mathtt{Beta}(\alpha,\beta)}{\partial\alpha}dx = \\
&= \frac{\partial \ln \mathtt{B}(\alpha,\beta)}{\partial\alpha} = \\
&= \frac{\mathrm{d}\ln\Gamma(\alpha)}{\mathrm{d}\alpha} - \frac{\partial\ln\Gamma(\alpha+\beta)}{\partial\alpha} = \\
&= \psi(\alpha) - \psi(\alpha+\beta)
\end{aligned}$$

□

Замечание 2.3. Отдельный интерес представляет подход к вычислению величины $w_{i,j} = \log\binom{d_{i,j}}{a_{i,j}}$. Дело в том, что числа с плавающей точкой в реализации большинства языков программирования имеют ограниченную точность, которая не позволяет отдельно вычислить биномиальный коэффициент, а потом взять от него логарифм. Даже в языке Python, в котором реализована длинная арифметика в целых числах, взятие логарифма может привести к переполнению типа при больших значениях $d_{i,j}$ и $a_{i,j} \simeq \frac{1}{2}d_{i,j}$.

Безусловно, можно вычислять $\log\binom{d_{i,j}}{a_{i,j}}$ как

$$\sum_{s=0}^{a_{i,j}-1} \log(d_{i,j}-s) - \sum_{s=0}^{a_{i,j}} \log s$$

Тем не менее, при большой разнице между $d_{i,j}$ и $a_{i,j}$ при таком подходе ошибки представления логарифмов накапливаются, что может привести к заметному расхождению подсчитанной величины с истинным значением.

Этих недостатков лишена красивая аппроксимация величины $\log n!$, полученная логарифмированием асимптотической формулы для $n!$, полученной великим математиком Сринивасой Рамануджаном[25] и уточняющей формулу Стирлинга.

$$\log n! \simeq n \log n - n + \frac{\log(n(1 + 4n(1 + 2n)))}{6} + \frac{\log \pi}{2}$$

Именно эта формула используется при вычислении $w_{i,j}$ в алгоритме XClone.

2.5.3.4 Вывод ELBO для RDR-модуля

Утверждение 2.5. Лог-обоснованность RDR-модуля можно записать как

$$\begin{aligned} &\underline{\mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma}} \log p(\boldsymbol{RD} \mid \boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma})} = \\ &= \int q(\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma}) \log p(\boldsymbol{RD} \mid \boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma}) \mathrm{d}\boldsymbol{m}, \mathrm{d}\boldsymbol{\gamma} = \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \widetilde{\pi}_{j,k} \cdot r_{ij} \cdot \mathbb{E}_{\boldsymbol{m},\boldsymbol{\gamma}} \log \widetilde{f}_{i,k} + C \end{aligned} \tag{2.24}$$

где $\widetilde{\cdot}$ указывает, что имеется в виду аппроксимация истинной апостериорной вероятности, а

$$\widetilde{f}_{i,k} \mid \boldsymbol{m},\boldsymbol{\gamma} := \frac{\sum_{t=1}^{T} m_i \exp[\gamma_t] \tilde{u}_{i,k,t}}{\sum_{b=1}^{N} \sum_{t=1}^{T} m_b \exp[\gamma_t] \tilde{u}_{b,k,t}}$$

Доказательство.

$$\int q(\boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{m}, \boldsymbol{\gamma}) \log p(\boldsymbol{RD} \mid \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{m}, \boldsymbol{\gamma}) \mathrm{d}\boldsymbol{m}, \mathrm{d}\boldsymbol{\gamma} =$$

$$= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma}} \log p(\boldsymbol{RD} \mid \boldsymbol{Z}, \boldsymbol{Y}, \boldsymbol{m}, \boldsymbol{\gamma}) =$$

$$= \text{см. формулу 2.16} =$$

$$= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma}} \left[\sum_{j=1}^{M} \sum_{k=1}^{K} \log p \left(\boldsymbol{r}_j \,\middle|\, \sum_{i=1}^{N} r_{i,j}, \boldsymbol{f}_k \right)^{z_{j,k}} \right] =$$

$$= \sum_{j=1}^{M} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma}} \left[z_{j,k} \cdot \log \left(r_j! \prod_{i=1}^{N} \frac{\widetilde{f}_{i,k}^{r_{i,j}}}{r_{i,j}!} \right) \right] =$$

$$= \text{ обозначим сумму всех констант через } C =$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{m},\boldsymbol{\gamma}} \left[z_{j,k} \cdot r_{ij} \log \widetilde{f}_{i,k} \right] + C =$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \widetilde{\pi}_{j,k} \cdot r_{ij} \cdot \mathbb{E}_{\boldsymbol{m},\boldsymbol{\gamma}} \log \widetilde{f}_{i,k} + C$$

□

2.5.4 Известные недостатки и планы по их исправлению

2.5.4.1 Избыточная сложность исходной модели

Эксперименты показали, что модель можно упростить:

- В коде XClone $\boldsymbol{m}$ фиксировано и выводится из теоретических соображений о равномерном распределении прочтений по геному. Т.е. $m_i = \frac{l_i}{\sum_{b=1}^{N} l_b}$ это достаточно хорошая оценка матожидания доли прочтений в сегменте i, где l_i это длина i-го блока в сегментации.

- Далее, среднее значение $\boldsymbol{\gamma}$ тоже фиксировано, $\mathbb{E}\gamma_t := \log \frac{c_{t,m}+c_{m,2}}{2} - \frac{e_2^{l_1^2}}{2}$, благодаря чему $\mathbb{E}\exp(\gamma_t) = \frac{c_{t,m}+c_{m,2}}{2}$, где l_1 это гиперпараметр ковариационной функции гауссовского процесса $\boldsymbol{\gamma}$. По изначальной задумке, по которой XClone можно было применять и для scRNA-seq данных, переменное $\boldsymbol{\gamma}$ было нужно для того, чтобы учесть не связанные

с числом копий причины аллельного-дисбаланса в транскриптомных данных.

После того, как стало ясно, что транскриптомные данные этой же моделью обработать не получится из-за шумного BAF-сигнала и более сложной природы RDR-сигнала (в силу разреженности данных), потребность в переменном $\boldsymbol{\gamma}$ отпала. Более того, в силу того, что соотв. KL-слагаемое давало несущественный вклад в функцию ошибки, распределение на $\boldsymbol{\gamma}$ буквально за несколько итераций вырождалось и теряло содержательный смысл.

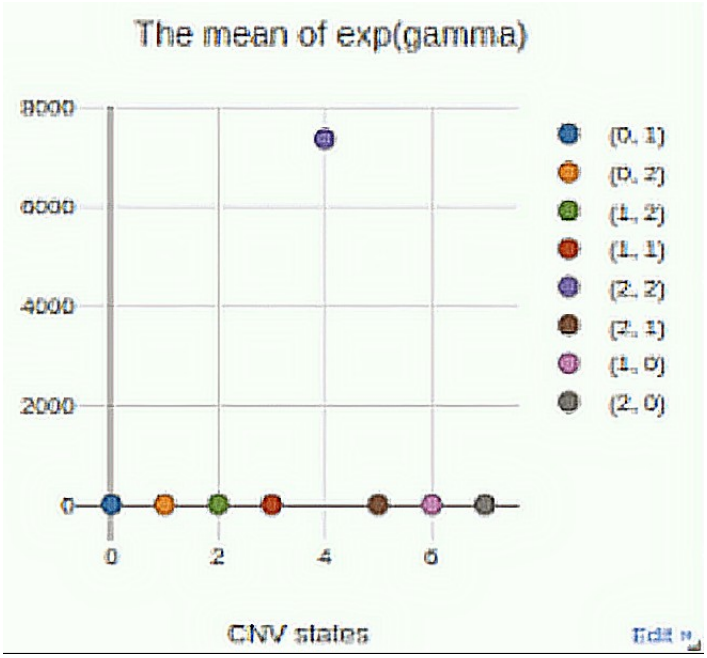


Рис. 2.10: Вырождение апостериорного распределения. Ось Ox — координаты, ось Oy — матожидание вдоль компоненты. До того, как было принято решение зафиксировать γ_t, распределение на $\boldsymbol{\gamma}$ буквально за пару итераций VB вырождалось в одномерное.

Такое поведение можно было подавить умножением соотв. KL-слагаемого на большой множитель (порядка 10^6), но такое преобразование было лишено смысла.

- Аналогично, связь между BAF- и RDR-модулями через $\boldsymbol{Z}$ и $\boldsymbol{Y}$ оказалась слишком слабой. Из-за этого получалась абсурдная ситуа-

ция, когда BAF-модуль выводил правильные θ, но они были никак не согласованы с ASCNV, которые выводил RDR-модуль. Как следствие было решено избавиться от первой размерности: $\forall i, j\ \forall t : \theta_{i,t} \overset{d}{=} \theta_{j,t}$, $\mathbb{E}\theta_{i,t} = \frac{c_{t,m}}{c_{t,m}+c_{t,p}}$, где $\overset{d}{=}$ означает равенство по распределению, и зафиксировать параметры α_t и β_t.

В самом деле, блок-зависимые BAF-ы имели смысл для транскриптомных данных, где картина экспрессии зависела от состояния клетки в момент секвенирования, что искажало сигнал аллельного дисбаланса. После того, как стало ясно, что VB будет применяться только к геномным данным, потребность в блок-специфичных BAF-ах отпала сама собой.

Более того, невооружённым взглядом видно, что BAF-модуль гораздо проще, чем RDR-модуль. Более того, гетерозиготные ОНП составляют малую долю от всего генома, и примерно половины прочтений теряется при переходе от ***RD*** к ***DP***. Как следствие, обоснованность RDR-модуля была на несколько порядков больше обоснованности BAF-модуля, да и оптимизировать её было значительно трудней. BAF-модуль быстро переобучался, буквально за несколько десятков итераций, после чего мешал VB находить оптимальные параметры RDR-модуля.

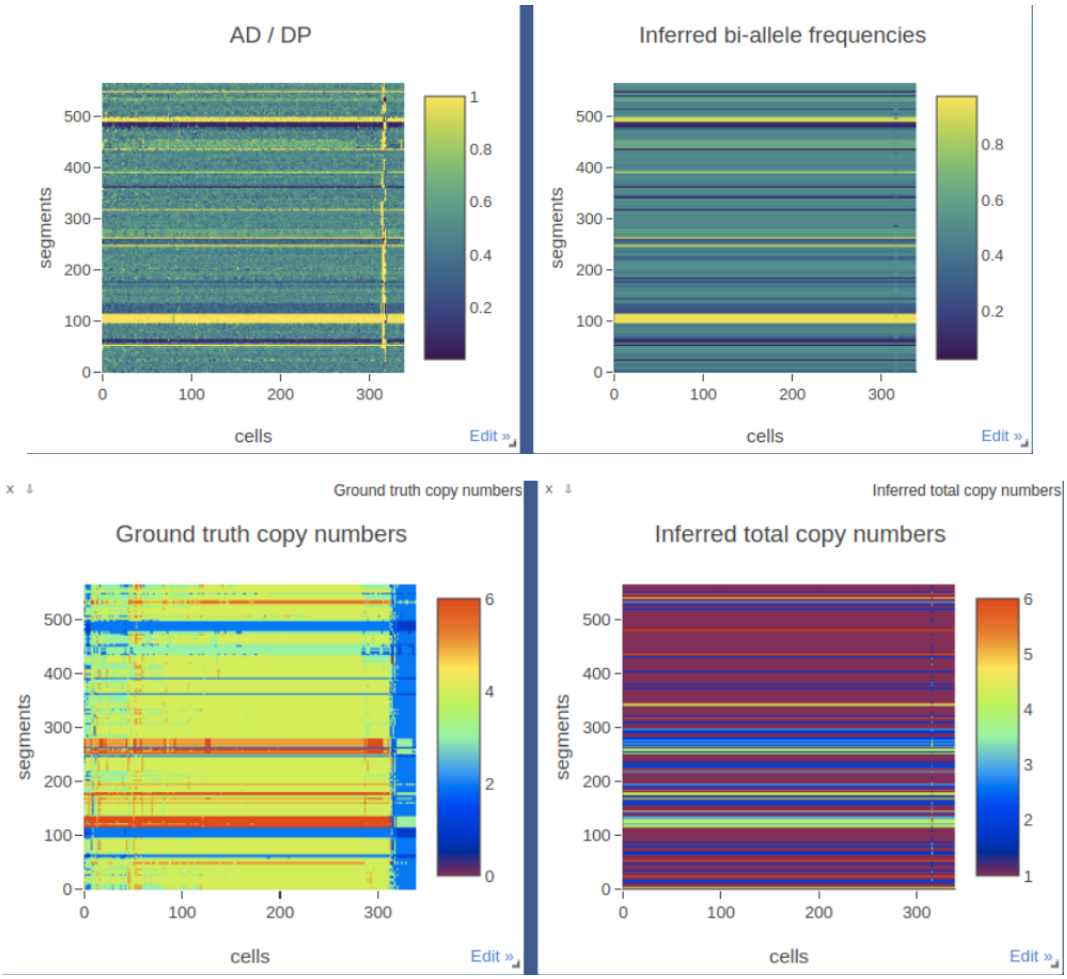


Рис. 2.11: Несогласованность найденных VB параметров при запуске на реальных данных при использовании блок-зависимых BAF-ов (на верхних тепловых картах). Слева — ожидаемые значения, справа — предсказанные. Видно, как BAF-модуль переобучается, а RDR-модуль хоть и улавливает какие-то закономерности, но, в целом, предсказывает что-то не то (даже по модулю того, что XClone не умеет детектировать whole-genome duplication).

Но как только частоты аллелей в блоках стали, по сути, определяться исключительно расстановкой наиболее вероятных Y и Z, BAF-модуль перестал переобучаться на шум в данных и изменения параметров модулей стали согласованными.

2.5.4.2 Численное интегрирование в оценке обоснованности RDR-модуля

Аналитическую формулу для обоснованости RDR-модуля, увы,

получить не удалось. Даже если фиксировать $\boldsymbol{m}$, неясно, как устроено отношение сумм зависимых логнормальных распределений в $\tilde{f}_{i,k}$. В [8] показано, что даже распределение числителя и знаменателя в отдельности имеет сложную структуру и не выражается через элементарные операции над стандартными распределениями. Как следствие, в коде алгоритма RDR-обоснованность приближается интегрированием по Монте-Карло.

Это существенный недостаток XClone в текущей редакции, т.к. метод Монте-Карло в общем случае имеет корневую сходимость[8], и для получения точности хотя бы до третьего знака приходится использовать порядка 10^6 точек, и это при фиксированном $\boldsymbol{m}$. В противном случае, пришлось бы семплировать ещё больше точек, чтобы сделать поправку на размерность пространства признаков (от нескольких сотен до нескольких тысяч).

2.5.4.3 Слишком большая разница масштабов отдельных слагаемых в ELBO

[8] Конспект лекции в MIT: https://ocw.mit.edu/courses/mechanical-engineering/2-086-numerical-computation-for-mechanical-engineers-fall-2014/nutshells-guis/MIT2_086F14_Monte_Carlo.pdf

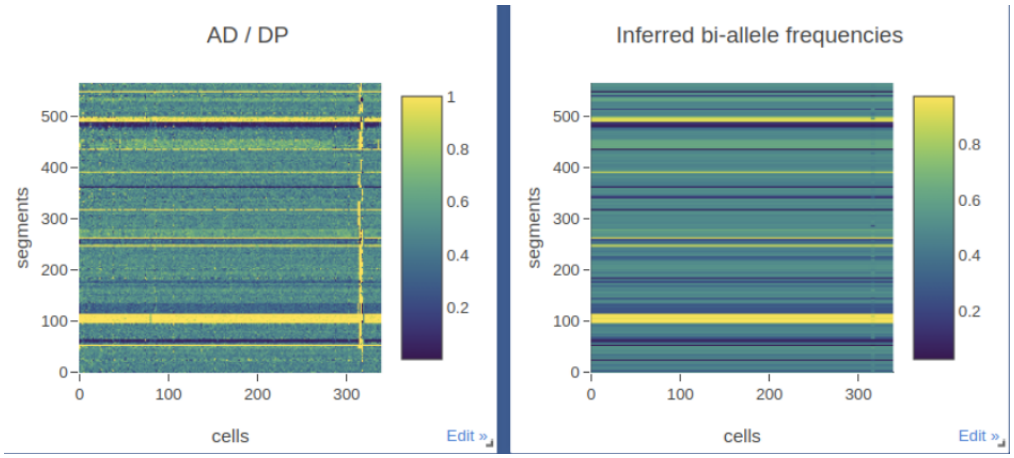


Рис. 2.12: Несогласованность найденных VB параметров при запуске на реальных данных при использовании блок-зависимых BAF-ов (на верхних тепловых картах). Слева — ожидаемые значения, справа — предсказанные. Видно, как BAF-модуль переобучается, а RDR-модуль хоть и улавливает какие-то закономерности, но, в целом, предсказывает что-то не то (даже по модулю того, что XClone не умеет детектировать whole-genome duplication).

2.5.4.4 Концептуальная невозможность детектирования WGD

АЛАЛАЛАЛАЛАЛА

2.5.4.5 Необходимость вручную задавать ожидаемое число клональных линий в образце

АЛАЛАЛАЛАЛАЛА

2.5.4.6 Отсутствие коррекции технических факторов в RDR-модуле

АЛАЛАЛАЛАЛАЛА

2.5.4.7 Предположение о независимости клональных линий. Игнорирование субклональной структуры

АЛАЛАЛАЛАЛАЛА

2.5.4.8 Наивный подход к определению ожидаемого числа прочтений в блоках сегментации

АЛАЛАЛАЛАЛАЛА

2.5.4.9 Практические трудности вычисления плотности биномиального распределения при большой глубине покрытия

АЛАЛАЛАЛАЛАЛА

Список использованных источников

[1] Richa Bharti и Dominik Grimm. «Current challenges and best-practice protocols for microbiome analysis». В: *Briefings in bioinformatics* (дек. 2019). DOI: 10.1093/bib/bbz155.

[2] David M. Blei, Alp Kucukelbir и Jon D. McAuliffe. «Variational Inference: A Review for Statisticians». В: *Journal of the American Statistical Association* 112.518 (февр. 2017), с. 859—877. ISSN: 1537-274X. DOI: 10.1080/01621459.2017.1285773. URL: http://dx.doi.org/10.1080/01621459.2017.1285773.

[3] Scott L. Carter и др. «Absolute quantification of somatic DNA alterations in human cancer». В: *Nature Biotechnology* 30.5 (май 2012), с. 413—421. ISSN: 1546-1696. DOI: 10.1038/nbt.2203. URL: https://doi.org/10.1038/nbt.2203.

[4] Y. Choi и др. «Comparison of phasing strategies for whole human genomes». В: *PLOS GENETICS* (апр. 2018). DOI: 10.1371/journal.pgen.1007308. URL: https://doi.org/10.1371/journal.pgen.1007308.

[5] Haplotype Reference Consortium. «A reference panel of 64,976 haplotypes for genotype imputation». В: *Nature Genetics* (48 авг. 2016), с. 1279—1283. DOI: 10.1038/ng.3643. URL: https://doi.org/10.1038/ng.3643.

[6] Stefan C. Dentro и др. «Portraits of genetic intra-tumour heterogeneity and subclonal selection across cancer types». В: *bioRxiv* (2018). DOI: 10.1101/312041. eprint: https://www.biorxiv.org/content/early/2018/07/13/312041.full.pdf. URL: https://www.biorxiv.org/content/early/2018/07/13/312041.

[7] Joshua V. Dillon и др. «TensorFlow Distributions». В: *CoRR* abs/1711.10604 (2017). arXiv: 1711.10604. URL: http://arxiv.org/abs/1711.10604.

[8] Daniel Dufresne. «The log-normal approximation in financial and other computations». В: *Advances in Applied Probability* 36 (3 июль 2004), с. 747—773. DOI: 10.1239/aap/1093962232. URL: https://doi.org/10.1239/aap/1093962232.

[9] Moritz Gerstung и др. «The evolutionary history of 2,658 cancers». В: *Nature* 578.7793 (февр. 2020), с. 122—128. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1907-7. URL: https://doi.org/10.1038/s41586-019-1907-7.

[10] Serin A. Harmanci, A.O. Harmanci и X. Zhou. «CaSpER identifies and visualizes CNV events by integrative analysis of single-cell or bulk RNA-sequencing data.» В: *Nature Communications* 89 (нояб. 2020). DOI: 10.1038/s41467-019-13779-x. URL: https://doi.org/10.1038/s41467-019-13779-x.

[11] Daniel C. Koboldt и др. «The Next-Generation Sequencing Revolution and Its Impact on Genomics». В: *Cell* 155 (1 сент. 2013). DOI: 10.1016/j.cell.2013.09.006. URL: https://doi.org/10.1016/j.cell.2013.09.006.

[12] H. W. Kuhn. «The Hungarian method for the assignment problem». В: *Naval Research Logistics Quarterly* 2.1-2 (1955), с. 83—97. DOI: 10.1002/nav.3800020109. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

[13] Emma Laks и др. «Resource: Scalable whole genome sequencing of 40,000 single cells identifies stochastic aneuploidies, genome replication states and clonal repertoires». В: *bioRxiv* (2018). DOI: 10.1101/411058. eprint: https://www.biorxiv.org/content/early/2018/09/13/411058.full.pdf. URL: https://www.biorxiv.org/content/early/2018/09/13/411058.

[14] Jacqueline A. Langdon и др. «Combined genome-wide allelotyping and copy number analysis identify frequent genetic losses without copy number reduction in medulloblastoma». В: *Genes, Chromosomes and Cancer*

45.1 (2006), с. 47—60. DOI: 10.1002/gcc.20262. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/gcc.20262. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/gcc.20262.

[15] Pablo Lapunzina и David Monk. «The consequences of uniparental disomy and copy number neutral loss-of-heterozygosity during human development and cancer». В: *Biology of the Cell* 103.7 (2011), с. 303—317. DOI: 10.1042/BC20110013. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1042/BC20110013. URL: https://onlinelibrary.wiley.com/doi/abs/10.1042/BC20110013.

[16] H. Li и др. «The Sequence Alignment/Map format and SAMtools». В: *Bioinformatics* (авг. 2009), с. 2087—2079. DOI: 10.1093/bioinformatics/btp352. URL: https://doi.org/10.1093/bioinformatics/btp352.

[17] «Linnarsson, S., Teichmann, S.A. Single-cell genomics: coming of age.» В: *Genome Biology* 97 (17 2016). DOI: 10.1186/s13059-016-0960-x. URL: https://doi.org/10.1186/s13059-016-0960-x.

[18] P.R. Loh и др. «Reference-based phasing using the Haplotype Reference Consortium panel». В: *Nature Genetics* (48 окт. 2016), с. 1443—1448. DOI: 10.1038/ng.3679. URL: https://doi.org/10.1038/ng.3679.

[19] Iain C. Macaulay, Wilfried Haerty и Parveen Kumar. «G&T-seq: parallel sequencing of single-cell genomes and transcriptomes». В: *Nature Methods* 12.6 (июнь 2015), с. 519—522. ISSN: 1548-7105. DOI: 10.1038/nmeth.3370. URL: https://doi.org/10.1038/nmeth.3370.

[20] D.J. McCarthy, R. Rostom и Huang Y. «Cardelino: computational integration of somatic clonal substructure and single-cell transcriptomes.» В: *Nature Methods* 17 (май 2020), с. 414—421. DOI: 10.1038/s41592-020-0766-3. URL: https://doi.org/10.1038/s41592-020-0766-3.

[21] «Method of the Year 2013». В: *Nature Methods* (11 янв. 2014). DOI: 10.1038/nmeth.2801. URL: https://doi.org/10.1038/nmeth.2801.

[22] «Method of the Year 2019: Single-cell multimodal omics». В: *Nature Methods* (17 янв. 2020). DOI: 10.1038/s41592-019-0703-5. URL: https://doi.org/10.1038/s41592-019-0703-5.

[23] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 978-0-262-01802-9.

[24] Erin D. Pleasance, R. Keira Cheetham и Philip J. Stephens. «A comprehensive catalogue of somatic mutations from a human cancer genome». В: *Nature* 463.7278 (янв. 2010), с. 191—196. ISSN: 1476-4687. DOI: 10.1038/nature08658. URL: https://doi.org/10.1038/nature08658.

[25] S. Ramanujan. *The Lost Notebook and other Unpublished Papers*. Под ред. S. S. Raghavan S. and Rangachari. Narosa, New Dehli: publisher, 1987.

[26] Sam Thiagalingam и др. «Mechanisms underlying losses of heterozygosity in human colorectal cancers». В: *Proceedings of the National Academy of Sciences* 98.5 (2001), с. 2698—2702. ISSN: 0027-8424. DOI: 10.1073/pnas.051625398. eprint: https://www.pnas.org/content/98/5/2698.full.pdf. URL: https://www.pnas.org/content/98/5/2698.

[27] Nicola Waddell и др. «Whole genomes redefine the mutational landscape of pancreatic cancer». В: *Nature* 518.7540 (февр. 2015), с. 495—501. ISSN: 1476-4687. DOI: 10.1038/nature14169. URL: https://doi.org/10.1038/nature14169.

[28] Huang Yuanhua, J. McCarthy Davis и Oliver Stegle. «Vireo: Bayesian demultiplexing of pooled single-cell RNA-seq data without genotype reference». В: *Genome Biology* 20 (дек. 2019). DOI: 10.1186/s13059-019-1865-2. URL: https://doi.org/10.1186/s13059-019-1865-2.

[29] S. Zaccaria и B.J. Raphael. «Characterizing the allele- and haplotype-specific copy number landscape of cancer genomes at single-cell resolution with CHISEL». В: *bioRxiv* (нояб. 2019). DOI: 10.1101/837195. URL: https://doi.org/10.1101/837195.

[30] Travis I. Zack и др. «Pan-cancer patterns of somatic copy number alteration». В: *Nature Genetics* 45.10 (окт. 2013), с. 1134—1140. ISSN: 1546-1718. DOI: 10.1038/ng.2760. URL: https://doi.org/10.1038/ng.2760.

[31] Hamim Zafar и др. «SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data». В: 29 (нояб. 2019), с. 1847—1859. DOI: 10.1101/gr.243121.118. URL: https://doi.org/10.1101/gr.243121.118.

[32] Hans Zahn и др. «Scalable whole-genome single-cell library preparation without preamplification». В: *Nature Methods* 14.2 (февр. 2017), с. 167—173. ISSN: 1548-7105. DOI: 10.1038/nmeth.4140. URL: https://doi.org/10.1038/nmeth.4140.