

Глава 1

Исследование и построение решения

1.1 Математическая модель поиска повторов в биологических последовательностях

Формально **биологическая последовательность**:

$$X = (x_n)_{n=1}^N, \quad N \in \mathbb{N}, \quad x_n \in \{A, T, G, C\},$$

где $N = |X|$ - длина последовательности, A, T, G, C - обозначения нуклеотидов.

Введем обозначение. **Подпоследовательностью** $X|_i^k$ последовательности X называется часть последовательности X с элемента с номером i длиной k при условии $i + k \leq |X|$:

$$X|_i^k = \{x_i, x_{i+1}, \dots, x_{i+k-1}\}$$

Под **повтором** будем понимать пару последовательностей (X_1, X_2) для которых справедливо неравенство:

$$\rho(X_1, X_2) \leq \varepsilon, \quad |X_1| = |X_2| = K$$

где:

- $\rho(X_1, X_2)$ - расстояние редактирования, оценка близости последовательностей, Конкретный вид функции расстояния редактирования $\rho(X_1, X_2)$ **определяется алгоритмом**
- ε - задаваемая точность поиска, значение которой будет **зависеть от задаваемой функции расстояния редактирования**
- K - длина последовательностей

Пусть есть две последовательности $X = (x_n)_{n=1}^{N_x}$ и $Y = (y_n)_{n=1}^{N_y}$.

Под **задачей поиска повторов** будем понимать нахождение всех троек $\{i_x, i_y, k\}$, $i_x, i_y, k \in \mathbb{N}$, таких что:

$$\begin{aligned} i_x + k &\leq N_x \\ i_y + k &\leq N_y \\ (X|_{i_x}^k, Y|_{i_y}^k) &- \text{повтор длины } k \end{aligned}$$

То есть задача найти все такие подпоследовательности последовательностей X и Y , что эти подпоследовательности будут повторами.

1.2 Спектрально-аналитический метод поиска повторов

В работе исследуется спектральный метод поиска повторов в биологических последовательностях предложенный и разработанный коллективом кафедры математических методов прогнозирования ВМК МГУ и института математических проблем в биологии Российской академии наук (Пушино).

Метод разбивается на четыре основных этапа:

1. Получение профилей биологических последовательностей
2. Спектральное индексирование полученных профилей
3. Сравнение коэффициентов спектрального индексирования и построение гомологической матрицы
4. Анализ гомологической матрицы

1.2.1 Профили биологических последовательностей

Одной из главных особенностей рассматриваемого метода поиска повторов является то, что на первом этапе последовательность преобразуется из дискретной в непрерывную область. Это достигается построением т.н. профилей.

Под GC-профилем последовательности $X = (x_n)_{n=1}^{N_x}$ с окном w в дальнейшем будем понимать такую последовательность $P_{GC}(X, w) = (p_n^{GC})_{n=1}^{N_p}$, $N_p = N_x - w + 1$, что

$$p_i^{GC} = \sum_{k=i}^{i+w} I^{GC}(x_k), i = \overline{1, N_p}$$

где

$$I^{GC} = \begin{cases} 1, & x_n \in \{G, C\} \\ 0, & \text{иначе} \end{cases}, n = \overline{1, N_x}$$

Понятие GA-профиля определяется аналогично:

$$P_{GA}(X, w) = (p_n^{GA})_{n=1}^{N_p} : p_i^{GA} = \sum_{k=i}^{i+w} I^{GA}(x_k), i = \overline{1, N_p},$$

где

$$I^{GA} = \begin{cases} 1, & x_n \in \{G, C\} \\ 0, & \text{иначе} \end{cases}, n = \overline{1, N_x}$$

1.2.2 Спектральное индексирование профилей

На этом этапе профили переводятся в спектральное представление с использованием в качестве базиса полиномов Чебышева дискретного аргумента.

Под спектральным представлением сигнала $P = (p_n)_{n=1}^{N_p}$ будем понимать вектор $\overline{C} = C_m(P) = (c_0, \dots, c_{m-1})$, где c_0, \dots, c_{m-1} - первые m коэффициентов разложения сигнала P по некоторой системе ортогональных функций $u_0(x), \dots, u_{m-1}(x), \dots$

Полиномы Чебышева определяются при помощи рекуррентного соотношения:

$$\begin{aligned}u_0(x) &= 1 \\u_1(x) &= x \\&\dots \\u_{n+1}(x) &= 2xu_n(x) - u_{n-1}(x)\end{aligned}$$

Вычисление коэффициентов разложения выполняется по рекуррентным соотношениям.

1.2.3 Спектральное сравнение профилей

На этой стадии спектральное представление профилей используется для производства сравнения на основе некоторого специально разработанного критерия.

Теперь можно ввести понятие расстояния редактирования для рассматриваемого метода поиска повторов.

Пусть X_1, X_2 - две биологические последовательности, такие что $|X_1| = |X_2| = K \geq w$. Будем понимать под GC- и GA-расстоянием редактирования для X_1 и X_2 :

$$\begin{aligned}\rho^{GC}(X_1, X_2) &= \left\| \overline{C}_1^{GC} - \overline{C}_2^{GC} \right\|, \\ \rho^{GA}(X_1, X_2) &= \left\| \overline{C}_1^{GA} - \overline{C}_2^{GA} \right\|, \\ \|\overline{C}\| &= \sum_{i=0}^{m-1} c_i^2\end{aligned}$$

Под повтором будем понимать пару последовательностей X_1, X_2 , удовлетворяющих системе:

$$\begin{cases} \rho^{GC}(X_1, X_2) < \varepsilon \\ \rho^{GA}(X_1, X_2) < \varepsilon \end{cases}$$

Под матрицей гомологии окном профилей w , окном аппроксимации a и шагом аппроксимации s для последовательностей X_1, X_2 будем понимать матрицу

$$M(X, Y, w, s, a) = (m_{ij})^{L_x \times L_y},$$

$$L_x = \lceil \frac{N_x - a - w + 1}{s} \rceil, L_y = \lceil \frac{N_y - a - w + 1}{s} \rceil$$

$$m_{ij} = \begin{cases} 1, & x_n \in \{G, C\} \\ 0, & \text{иначе} \end{cases}$$

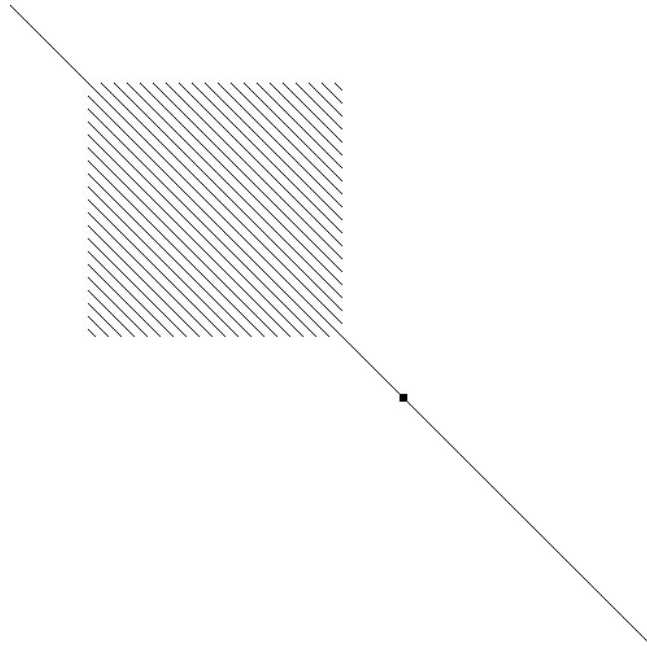


Рисунок 1.1: Пример гомологической матрицы

На рисунке 1.1 приведена гомологическая матрица, образованная в результате сравнения искусственной последовательности с самой собой. Матрица имеет симметричный вид.

1.2.4 Анализ гомологической матрицы

Рассмотрим все диагонали гомологической матрицы параллельные главной диагонали, включая ее саму. Фиксируем диагональ D_l . Назовем диагональным элементом $R(x, y, k)$ такое множество элементов m_{ij} гомологической матрицы $M^{L_x \times L_y}$, что:

$$m_{ij} \in D_l$$

$$m_{ij} = 1, \quad i = \overline{y, y+k-1}, \quad j = \overline{x, x+k-1}, k \in \mathbb{N}$$

$$m_{x-1, y-1} = 0, \text{ если } x > 0, y > 0$$

$$m_{x+k, y+k} = 0, \text{ если } x+k < L_x, y+k < L_y$$

На последнем этапе анализа гомологической матрицы нужно найти все диагональные элементы этой матрицы. Зная параметры алгоритма по диагональным элементам можно восстановить искомые повторы.

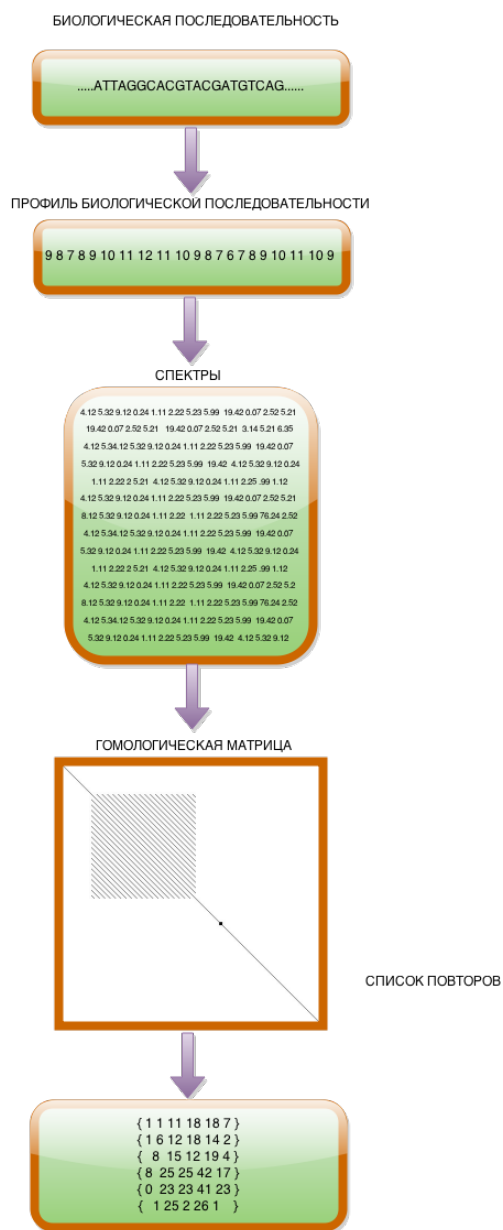


Рисунок 1.2: Схема работы спектрально-аналитического метода поиска повторов

Но такой подход имеет и ряд недостатков: дублирование вычислений, сложность отладки, невозможность повторного использования данных для других параметров алгоритма.

К недостаткам предложенного алгоритма можно отнести обмены между процессами. Однако каждый процесс может начинать работу не дожидаясь завершения приема данных, т.к. они понадобятся только в конце работы. Таким образом можно скрыть передчу данных на фоне выполнения полезной работы и добиться ситуации, когда процессор не будет простаивать в ожидании данных.

Ниже рассматриваются два способа организации этапа спектрального сравнения и анализа гомологической матрицы.

1.3.2 Спектральное сравнение и анализ матрицы гомологии. Матричный метод

Во время спектрального сравнения каждый процесс будет отвечать за сравнение своей части спектров первой последовательности со всеми спектрами второй последовательности. Таким образом процесс будет хранить часть строк гомологической матрицы, соответствующих спектрам из первого набора.

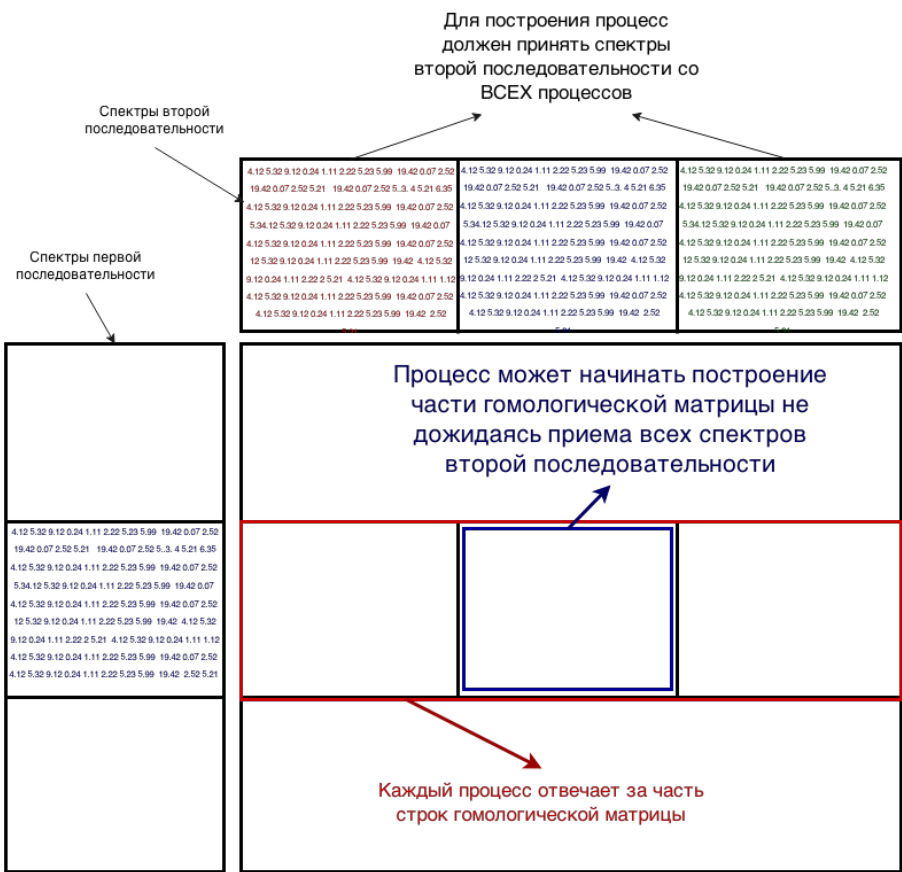


Рисунок 1.4: Сравнение спектров. Матричный метод

Для выполнения данного этапа процессу требуются собрать спектры второй последовательности со всех процессов. Однако и здесь можно добиться отсутствия простоя вычислительных платформ, если не дожидаться получения всех данных, а начинать работу по возможности. То есть как только придут данные хотя бы с одного процесса - сразу строить часть матрицы. Таким образом можно скрыть большую часть пересылок данных за полезными вычислениями и добиться хорошей масштабируемости алгоритма.

На последнем этапе каждый процесс анализирует имеющуюся часть гомологической матрицы на наличие диагональных элементов. Однако следует учитывать, что найденный диагональный элемент может иметь свое начало или окончание на другом процессе и таким образом быть разбитым по процессам, поэтому нужно "склеить" диагональные элементы с разных процессов.

Здесь будем действовать по следующему принципу: процесс "склеивает" диагональный элемент (т.е. ищет его целиком), только если диагональный элемент принадлежит нашему процессу. Диагональный элемент $R(x, y, k)$ принадлежит процессу i , если начало этого повторения находится на процессе i , т.е.:

$$R(x, y, k) \in i, \text{ если } m_{xy} \in M_i^{L_x \times L_y},$$

где $M_i^{L_x \times L_y}$ — гомологическая матрица, вычисленная на процессе i

Таким образом алгоритм анализа гомологической матрицы будет работать следующим образом:

1. Найти все диагональные элементы
2. Удалить диагональные элементы, не принадлежащие нашему процессу
3. Найти конец всех диагональных элементов

Такой алгоритм нахождения диагональных элементов, при котором последовательно идут этапы спектрального сравнения с полным построением матрицы гомологии и ее последующим анализом будем называть матричным методом.

1.3.3 Спектральное сравнение и анализ матрицы гомологии. Блочный метод

Явным недостатком матричного метода является объем памяти, необходимый для гомологической матрицы. Размер этой матрицы $\sim N/s$, где N - длина входной последовательности, s - сдвиг окна индексирования. Тогда, например, объем гомологической матрицы, необходимый для анализа человеческого генома (3.1 млрд. н.п.) со сдвигом s равным 100, будет составлять $(3.1 * 10^7 * 3.1 * 10^7 \sim 9 * 10^{14})$ 900 ТБ. Такой размер гомологической матрицы существенно сокращает количество возможных вычислительных платформ на которых можно было бы запускать программу.

Чтобы уйти от этого недостатка нам придется пренебречь принципом автономности этапов. Основная идея блочного метода заключается в чередовании этапов спектрального индексирования и анализа гомологической матрицы для обработки небольшого блока.

Итак, пусть пользователь задает новый параметр алгоритма - лимит оперативной памяти для гомологической матрицы. Также как и в матричном методе процесс будет отвечать за сравнение спектров первой последовательности со всеми спектрами второй последовательности. Единственное отличие блочного метода от матричного в том, что результат сравнения не будет сохраняться в матрице гомологии, а будет сразу же подвергаться анализу на наличие диагональных элементов. Таким образом гомологическая матрица становится чисто виртуальной структурой - на деле ничего в памяти не хранится.

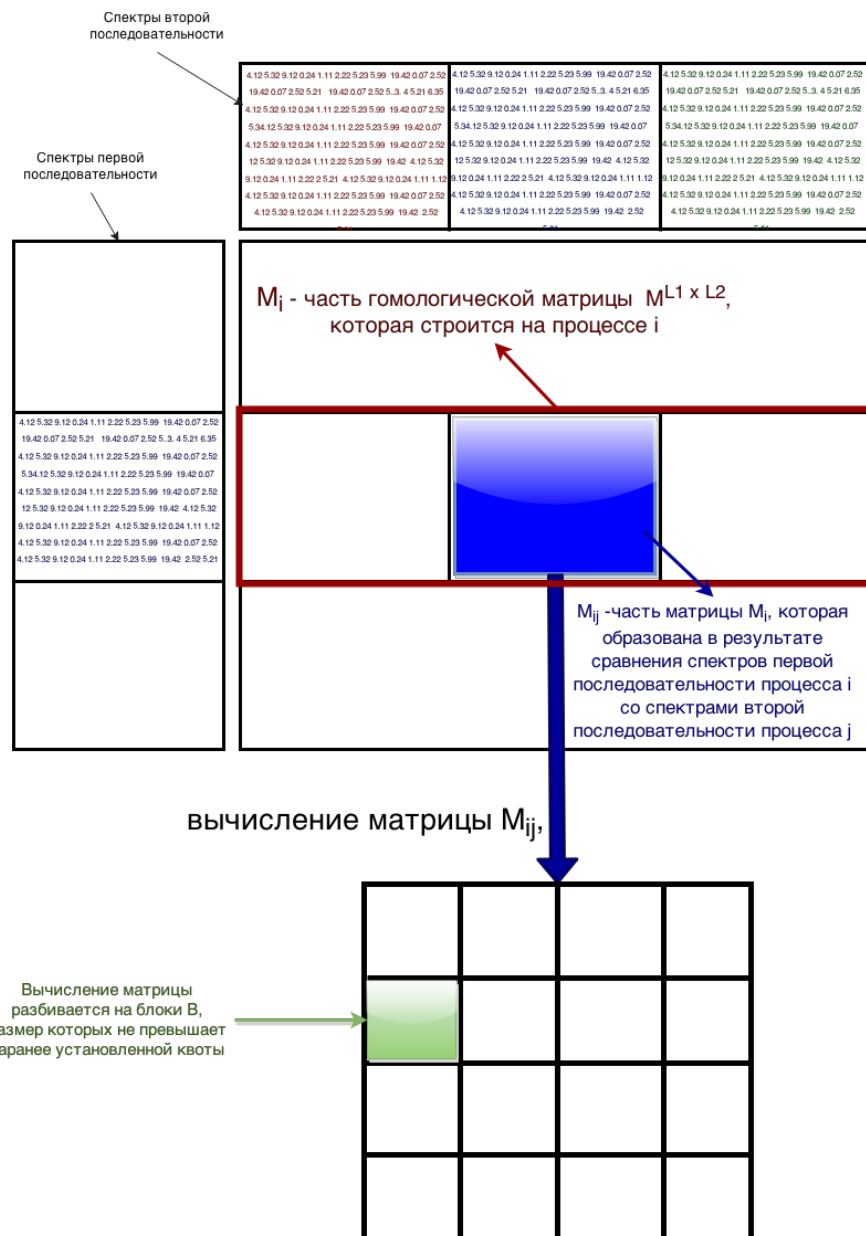


Рисунок 1.5: Сравнение спектров. Блочный метод

На рис. 1.5 разобрана схема блочного метода. Как только процесс принял часть спектров второй последовательности начинается их сравнение со спектрами первой последовательности. Выделяется блок памяти, объем которого не превышает заранее определенной квоты пользователя. Строится часть виртуальной гомологической матрицы равная размеру выделенного блока. Блок сразу же подвергается анализу и результат анализа сохраняется в списке диагональных элементов.

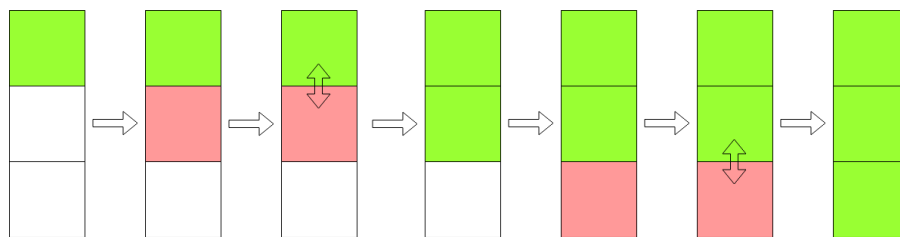


Рисунок 1.6: Склейка по вертикали

Блок сдвигается ниже на его высоту, идет построение и анализ. Результат анализа “склеивается” по вертикали с предыдущими результатами (рис. 1.6). Операция "склейки" диагональных элементов по вертикали не имеет ничего общего со "склежкой" между процессами, т.к. здесь процесс знает всю информацию о склеиваемых структурах.



Рисунок 1.7: Склейка по горизонтали

После построения столбца виртуальной гомологической матрицы результат сохраняется и происходит построение следующего столбца. После его постройки происходит “склейка” диагональных элементов по горизонтали. И так далее, пока не получим результат анализа с одним из процессов (см. рис 1.7).

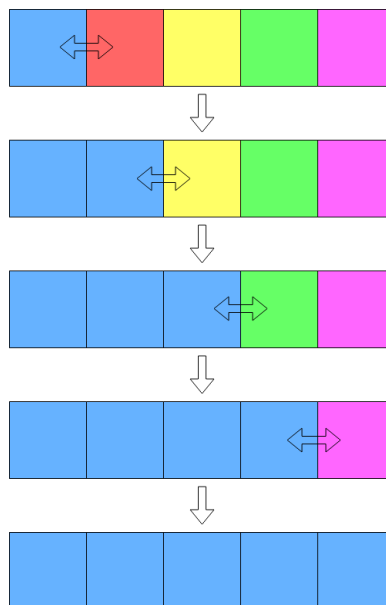


Рисунок 1.8: Склейка по горизонтали процессов

На последнем этапе будем иметь результаты анализа сравнения со всеми процессами. Остается применить ко всем операцию “склейка” горизонтали, и получим список диагональных элементов (см. рис 1.8).

То есть будем иметь такой же список, как и при матричном способе. Остается применить операцию “склейка” процессам и перевести диагональные элементы в искомые повторы.

Список литературы

- [1] Антонов А. С. Технологии параллельного программирования MPI и OpenMP. — Издательство Московского университета, 2012.
- [2] Колесин М. С. — Исследование эффективности реализации спектрального метода поиска неточных повторов в последовательностях ДНК на графических процессорах. — Master's thesis, МГУ ВМК, 2011.
- [3] Поиск протяженных повторов в геномах на основе спектрально-аналитического метода / А. Н. Панкратов, М. И. Пятков, Р. К. Тетуев et al. // *Математическая биология и биоинформатика*. — 2012. — Vol. 7, no. 2. — Pp. 476–492.
- [4] Пятков М. И. Распараллеливание спектрального алгоритма поиска повторов в геномных последовательностях // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи: Труды Международной суперкомпьютерной конференции (20-25 сентября 2010 г., г. Новороссийск). — Издательство Московского университета, 2010. — Pp. 663–666.
- [5] Пятков М. И. Спектрально-аналитический метод поиска протяженных повторяющихся нуклеотидных последовательностей в геномах: Ph.D. thesis / Институт математических проблем биологии Российской академии наук. — 2013.
- [6] Распараллеливание спектрального алгоритма поиска повторов в геномах / А. Н. Панкратов, М. И. Пятков, С. А. Комаров et al. // Научный сервис в сети Интернет: экзафлопсное будущее: Труды Международной суперкомпьютерной конференции (19-24 сентября 2011 г., г.Новороссийск). — Издательство Московского университета, 2011. — Pp. 304–306.
- [7] Сандерс Д., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров / Под ред. А. В. Боресков. — ДМК Пресс, 2013.