

Московский Государственный Университет им. М. В. Ломоносова  
Факультет Вычислительной Математики и Кибернетики  
Кафедра Суперкомпьютеров и Квантовой Информатики

Тойгильдин Владислав

**Разработка и исследование  
параллельного алгоритма поиска  
неточных повторов в геноме.**

Научный руководитель:  
к.ф.-м.н., доцент  
Попова Нина Николаевна

апрель 2015

# Цель работы

Дипломная работа посвящена параллельным методам анализа и обработки биологических последовательностей на суперкомпьютерах с целью сокращения времени обработки и увеличения объема обрабатываемых данных.

# Актуальность работы

В работе рассматривается задача поиска неточных протяженных повторов в биологических последовательностях. Важность поиска повторяющихся элементов обусловлена биологической ролью повторов в функционировании организмов.

Нахождение повторов позволяет решать многие задачи:

- Определение родства групп организмов на генном уровне.
- Диагностика генетических заболеваний.
- и другие

# Мотивация постановки задачи

Мотивация постановки задачи связана с растущими объемами генетических данных и необходимостью проведения многократных быстрых вычислительных экспериментов.

В данной работе предлагается и исследуется возможность суперкомпьютерной реализации метода поиска повторов в биологических последовательностях.

# Рассматриваемый метод

В дипломной работе рассматривается метод поиска протяженных размытых повторов, разработанный коллективом сотрудников кафедры математических методов прогнозирования ВМК МГУ и института математических проблем в биологии Российской академии наук (Пушино).

Дипломная работа выполнялась в тесном сотрудничестве с авторами метода.<sup>1</sup>

---

<sup>1</sup>Тетуев .Р.К., Назипова Н.Н., Панкратов А.Н., Дедус Ф.Ф. Поиск мегасателлитных повторов в геномах эукариот по оценке осцилляций кривых GC- содержания// Математическая биология и биоинформатика, 2010, Т. 5, № 1.

# Задачи дипломной работы

- Разработка и реализация параллельного алгоритма поиска повторов, ориентированного на суперкомпьютерную реализацию: большой объем обрабатываемых данных (размер данных порядка 1 ГБ), время обработки в пределах 1 часа.
- Исследование эффективности использования графических процессоров для решения поставленной задачи.
- Анализ масштабируемости разработанного алгоритма на примере задачи сравнения конкретных геномов
- Разработка графического интерфейса для работы пользователя на локальной и удаленных системах.

# Вычислительные платформы

Вычислительные платформы для решения поставленных задач:

- суперкомпьютеры Ломоносов, BlueGene/P
- 2-х процессорная рабочая станция на базе 4-х ядерного процессора Intel Xeon E5630 и 4-х графических ускорителей Tesla K40, Tesla K20c, 2xTesla C2075

Предложенные платформы обладают различной архитектурой и позволяют исследовать разрабатываемый алгоритм с учетом их особенностей.

# Исходные данные

Объектом исследования являются биологические последовательности. Длины исследуемых последовательностей достигают порядка  $10^6 - 10^9$  элементов.

Источником используемых данных являются всемирные базы данных генетической информации, например база данных национального центра биотехнической информации США (NCBI).<sup>2</sup>

---

<sup>2</sup><http://www.ncbi.nlm.nih.gov/>



# Исследуемые последовательности

Необходимо найти **протяженные неточные** повторы.

- протяженные - длинные повторы (от 1000 эл.)
- неточные - следствие воздействия мутационных процессов (удаление, вставка, делеция)

# Математическая модель

Формально **биологическая последовательность**:

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

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

Для последовательности  $X$  под  $X|_i^k$  при условии  $i + k < |X|$  будем понимать **часть последовательности  $X$  с элемента с номером  $i$  длиной  $k$** :

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

# Математическая модель

Под **повтором** будем понимать пару последовательностей  $(X, Y)$ , для которых справедливо неравенство:

$$\rho(X, Y) \leq \varepsilon, \quad |X| = |Y| = K$$

где:

- $K$  - длина последовательностей,
- $\rho(X, Y)$  - т.н. расстояние редактирования, оценка близости последовательностей, [конкретный вид функции расстояния редактирования  $\rho(X, Y)$  **определяется алгоритмом**],
- $\varepsilon$  - задаваемая точность поиска, значение которой будет зависеть от задаваемой функции расстояния.

# Математическая модель

Пусть есть две биологические последовательности

$$X = (x_n)_{n=1}^{N_x} \text{ и } Y = (y_n)_{n=1}^{N_y}$$

Под **задачей поиска повторов** будем понимать нахождение

всех троек  $\{i_x, i_y, k\}$ ,  $i_x, i_y, k \in \mathbb{N}$ , таких что:

$i_x + k \leq N_x, i_y + k \leq N_y$  и  $(X|_{i_x}^k, Y|_{i_y}^k)$  - повтор длины  $k$ , т.е.:

$$\rho(X|_{i_x}^k, Y|_{i_y}^k) \leq \varepsilon$$



# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA
```

# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACAGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA
```

L1 - длина окна; параметр алгоритма

## Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

GTAGCATGGTAGAGATGTAGCACA GCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA

считаем в окне некую  
статистическую величину и получаем  
неотрицательное число

4



# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
CTAGCATGGTAGAGATGTAGCACAATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA
```



сдвигаем окно на 1 символ

# Спектрально-аналитический метод

`char sequence[N]; // A, T, G, C`

`CTAGCATGGTAGAGATGTAGCACAATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA`



45

# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
CTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCTACGTAGCATCGATGCATGCGTATAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA
```

456

# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA
```



из последовательности символов получаем  
дискретную числовую функцию

```
45676567898789887876545654654321210101012345454345676567898765656765676787
```



# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTCGTCGACGATA
```

```
unsigned int profile[N - L1 + 1];
```

45676567898789887876545654654321210101012345454345676567898765656765676787

L2 - длина окна; параметр алгоритма

# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA
```

```
unsigned int profile[N - L1 + 1];
```

```
45676567898789887876545654654321210101012345454345676567898765656765676787
```

↓ считаем спектры - первые  $m$  коэффициентов разложения

1.42  
3.23  
2.12  
6.23  
4.76  
2.74  
4.02  
1.63  
6.35  
4.14  
7.52  
0.25  
1.25  
0.67  
2.52  
1.25

}  $m$  - параметр алгоритма

# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCTACGTAGCATCGATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCAGTCGACGATA
```

```
unsigned int profile[N - L1 + 1];
```

45676567898789887876545654654321210101012345454345676567898765656765676787

сдвигаем окно на s элементов; s - параметр алгоритма

1.42  
3.23  
2.12  
6.23  
4.76  
2.74  
4.02  
1.63  
6.35  
4.14  
7.52  
0.25  
1.25  
0.67  
2.52  
1.25



# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGAGTCGAGTCGACGATA
```

```
unsigned int profile[N - L1 + 1];
```

45676567898789887876545654654321210101012345454345676567898765656765676787

↓

1.42	6.35
3.23	4.14
2.12	7.52
6.23	0.25
4.76	1.25
2.74	0.67
4.02	2.52
1.63	1.25
6.35	1.42
4.14	3.23
7.52	2.12
0.25	6.23
1.25	4.76
0.67	2.74
2.52	4.02
1.25	1.63

# Спектрально-аналитический метод

char sequence[N]; // A, T, G, C

GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCAGTCAGTCGGACGATA

unsigned int profile[N - L1 + 1];

45676567898789887876545654654321210101012345454345676567898765656765676787

↓

1.42	6.35	0.67
3.23	4.14	2.52
2.12	7.52	1.25
6.23	0.25	1.42
4.76	1.25	3.23
2.74	0.67	2.12
4.02	2.52	6.23
1.63	1.25	4.76
6.35	1.42	2.74
4.14	3.23	4.02
7.52	2.12	1.63
0.25	6.23	6.35
1.25	4.76	4.14
0.67	2.74	7.52
2.52	4.02	0.25
1.25	1.63	1.25

# Спектрально-аналитический метод

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCAGTCAGTCGCGACGATA
```

```
unsigned int profile[N - L1 + 1];
```

45676567898789887876545654654321210101012345454345676567898765656765676787

1.42	6.35	0.67
3.23	4.14	2.52
2.12	7.52	1.25
6.23	0.25	1.42
4.76	1.25	3.23
2.74	0.67	2.12
4.02	2.52	6.23
1.63	1.25	4.76
6.35	1.42	2.74
4.14	3.23	4.02
7.52	2.12	1.63
0.25	6.23	6.35
1.25	4.76	4.14
0.67	2.74	7.52
2.52	4.02	0.25
1.25	1.63	1.25

Продолжаем тоже самое  
со второй последовательностью



# Спектрально-аналитический метод

## Набор спектров первой последовательности

m - количество коэффициентов

6.27	9.87	7.55	9.63	8.67
7.72	2.00	9.83	0.60	3.84
9.76	5.04	8.77	7.18	9.64
5.59	4.32	1.35	5.91	2.34
8.87	8.58	4.14	5.24	9.31
8.52	0.74	5.75	1.69	5.16
6.73	7.25	6.21	2.93	9.58
6.70	5.33	9.32	3.69	4.22
3.63	2.07	6.30	3.89	0.75
9.33	9.80	8.53	5.18	8.12
6.72	3.21	0.11	1.53	7.08
6.79	8.49	5.09	6.68	2.78
8.04	4.90	8.69	3.84	4.89
8.62	6.79	3.67	4.36	0.69
0.96	2.16	6.33	4.25	6.23
0.43	9.14	2.11	8.77	1.69
7.91	5.28	2.14	1.59	4.16
7.94	9.75	4.58	7.35	9.09
5.85	7.86	4.87	2.81	5.91
7.30	4.81	8.52	1.53	3.46
2.38	0.82	1.51	8.22	4.22
2.45	6.85	5.05	0.27	5.35
2.36	10.00	5.23	4.12	8.62
2.46	8.98	3.14	8.67	9.35
8.68	2.95	2.10	7.24	5.87
8.55	5.24	5.40	7.07	7.24
7.10	6.70	2.97	6.58	1.21
4.51	7.72	5.42	5.47	2.19
5.25	9.57	9.30	0.97	4.49
2.06	7.91	7.68	8.60	2.51

N1 - количество спектров из первого набора

### Набор спектров второй последовательности

## Спектрально-аналитический метод

Для удобства транспонируем второй набор

**N1**

6.27	9.87	7.55	9.63	8.67
7.72	2.00	9.83	0.60	3.84
9.76	5.04	8.77	7.18	9.64
5.59	4.32	1.35	5.91	2.34
8.87	8.58	4.14	5.24	9.31
8.52	0.74	5.75	1.69	5.16
6.73	7.25	6.21	2.93	9.58
6.70	5.33	9.32	3.69	4.22
3.63	2.07	6.30	3.89	0.75
9.33	9.80	8.53	5.18	8.12
6.72	3.21	0.11	1.53	7.08
6.79	8.49	5.09	6.68	2.78
8.04	4.90	8.69	3.84	4.89
8.62	6.79	3.67	4.36	0.69
0.96	2.16	6.33	4.25	6.23
0.43	9.14	2.11	8.77	1.69
7.91	5.28	2.14	1.59	4.16
7.94	9.75	4.58	7.35	9.09
5.85	7.86	4.87	2.81	5.91
7.30	4.81	8.52	1.53	3.46
2.38	0.82	1.51	8.22	4.22
2.45	6.85	5.05	0.27	5.35
2.36	10.00	5.23	4.12	8.62
2.46	8.99	3.14	8.67	9.35
8.68	2.95	2.10	7.24	5.87
8.55	5.24	5.40	7.07	7.24
7.10	6.70	2.97	6.58	1.21
4.51	7.72	5.42	5.47	2.19
5.25	9.57	9.30	0.97	4.49
2.06	7.91	7.68	8.60	2.51

**N2**

8.23	3.46	5.61	2.32	4.56	3.52	2.88	3.64	6.09	5.95	4.63	1.22	6.94	6.46	5.94	5.03	8.63	0.50	5.86	0.64	4.95	6.35	6.35	8.21	8.57
7.62	4.12	0.60	8.66	5.56	1.97	6.15	4.49	7.22	4.70	7.10	3.58	2.42	1.70	4.22	2.97	9.96	9.21	4.22	0.75	3.87	2.53	5.32	8.26	3.13
0.71	3.35	5.21	0.73	8.58	7.18	0.83	0.02	6.57	7.77	3.27	7.88	1.72	2.61	6.01	3.82	5.41	2.84	6.54	8.00	7.47	3.93	4.48	5.74	6.95
9.18	7.27	1.78	4.07	6.57	6.95	0.90	8.10	5.25	8.98	2.20	2.83	7.18	0.69	9.68	7.63	5.30	6.82	3.52	3.18	9.38	8.61	3.26	8.80	0.17
4.66	2.92	5.48	5.07	1.75	8.57	6.78	8.93	3.35	0.44	3.72	3.67	8.85	6.34	5.23	5.39	7.38	4.06	9.04	2.25	9.23	3.23	6.63	0.03	4.51

# Спектрально-аналитический метод

```
8.23 3.46 5.61 2.32 4.56 3.52 2.88 3.64 6.09 5.95 4.63 1.22 6.94 6.46 5.94 5.03 8.63 0.50 5.86 0.64 4.95 6.35 6.35 8.21 8.57
7.62 4.12 0.60 8.66 5.56 1.97 6.15 4.49 7.22 4.70 7.10 3.58 2.42 1.70 4.22 2.97 9.96 9.21 4.22 0.75 3.87 2.53 5.32 8.26 3.13
0.71 3.35 5.21 0.73 8.58 7.18 0.83 0.02 6.57 7.77 3.27 7.88 1.72 2.61 6.01 3.82 5.41 2.84 6.54 8.00 7.47 3.93 4.48 5.74 6.95
9.18 7.27 1.78 4.07 6.57 6.95 0.90 8.10 5.25 8.98 2.20 2.83 7.18 0.69 9.68 7.63 5.30 6.82 3.52 3.18 9.38 8.61 3.26 8.80 0.17
4.66 2.92 5.48 5.07 1.75 8.57 6.78 8.93 3.35 0.44 3.72 3.67 8.85 6.34 5.23 5.39 7.38 4.06 9.04 2.25 9.23 3.23 6.63 0.03 4.51
```

```
6.27 9.87 7.55 9.63 8.67
7.72 2.00 9.83 0.60 3.84
9.76 5.04 8.77 7.18 9.64
5.59 4.32 1.35 5.91 2.34
8.87 8.58 4.14 5.24 9.31
8.52 0.74 5.75 1.69 5.16
6.73 7.25 6.21 2.93 9.58
6.70 5.33 9.32 3.69 4.22
3.63 2.07 6.30 3.89 0.75
9.33 9.80 8.53 5.18 8.12
6.72 3.21 0.11 1.53 7.08
6.79 8.49 5.09 6.68 2.78
8.04 4.90 8.69 3.84 4.89
8.62 6.79 3.67 4.36 0.69
0.96 2.16 6.33 4.25 6.23
0.43 9.14 2.11 8.77 1.69
7.91 5.28 2.14 1.59 4.16
7.94 9.75 4.58 7.35 9.09
5.85 7.86 4.87 2.81 5.91
7.30 4.81 8.52 1.53 3.46
2.38 0.82 1.51 8.22 4.22
2.45 6.85 5.05 0.27 5.35
2.36 10.00 5.23 4.12 8.62
2.46 8.98 3.14 8.67 9.35
8.68 2.95 2.10 7.24 5.87
8.55 5.24 5.40 7.07 7.24
7.10 6.70 2.97 6.58 1.21
4.51 7.72 5.42 5.47 2.19
5.25 9.57 9.30 0.97 4.49
2.06 7.91 7.68 8.60 2.51
```

Сравним каждый спектр из первого набора  
с каждым спектром из второго набора  
и построим гомологическую матрицу,  
отражающую результат сравнения двух наборов

```
bool matrixGomology[N1][N2];
```



# Спектрально-аналитический метод

8.23 3.46 5.61 2.32 4.56 3.52 2.88 3.64 6.09 5.95 4.63 1.22 6.94 6.46 5.94 5.03 8.63 0.50 5.86 0.64 4.95 6.35 6.35 8.21 8.57  
7.62 4.12 0.60 8.66 5.56 1.97 6.15 4.49 7.22 4.70 7.10 3.58 2.42 1.70 4.22 2.97 9.96 9.21 4.22 0.75 3.87 2.53 5.32 8.26 3.13  
0.71 3.35 5.21 0.73 8.58 7.18 0.83 0.02 6.57 7.77 3.27 7.88 1.72 2.61 6.01 3.82 5.41 2.84 6.54 8.00 7.47 3.93 4.48 5.74 6.95  
9.18 7.27 1.78 4.07 6.57 6.95 0.90 8.10 5.25 8.98 2.20 2.83 7.18 0.69 9.68 7.63 5.30 6.82 3.52 3.18 9.38 8.61 3.26 8.80 0.17  
4.66 2.92 5.48 5.07 1.75 8.57 6.78 8.93 3.35 0.44 3.72 3.67 8.85 6.34 5.23 5.39 7.38 4.06 9.04 2.25 9.23 3.23 6.63 0.03 4.51

6.27 9.87 7.55 9.63 8.67  
7.72 2.00 9.83 0.60 3.84  
9.76 5.04 8.77 7.18 9.64  
5.59 4.32 1.35 5.91 2.34  
8.87 8.58 4.14 5.24 9.31  
8.52 0.74 5.75 1.69 5.16  
6.73 7.25 6.21 2.93 9.58  
6.70 5.33 9.32 3.69 4.22  
3.63 2.07 6.30 3.89 0.75  
9.33 9.80 8.53 5.18 8.12  
6.72 3.21 0.11 1.53 7.08  
6.79 8.49 5.09 6.68 2.78  
8.04 4.90 8.69 3.84 4.89  
8.62 6.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.62  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24  
7.10 6.70 2.97 6.58 1.21  
4.51 7.72 5.42 5.47 2.19  
5.25 9.57 9.30 0.97 4.49  
2.06 7.91 7.68 8.60 2.51

Сравнение спектров проведем с помощью  
среднеквадратичного отклонения

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}.$$

$\sigma < \epsilon$ , где  $\epsilon$  - параметр алгоритма

# Спектрально-аналитический метод

8.23 3.46 5.61 2.32 4.56 3.52 2.88 3.64 6.09 5.95 4.63 1.22 6.94 6.46 5.94 5.03 8.63 0.50 5.86 0.64 4.95 6.35 6.35 8.21 8.57  
7.62 4.12 0.60 8.66 5.56 1.97 6.15 4.49 7.22 4.70 7.10 3.58 2.42 1.70 4.22 2.97 9.96 9.21 4.22 0.75 3.87 2.53 5.32 8.26 3.13  
0.71 3.35 5.21 0.73 8.58 7.18 0.83 0.02 6.57 7.77 3.27 7.88 1.72 2.61 6.01 3.82 5.41 2.84 6.54 8.00 7.47 3.93 4.48 5.74 6.95  
9.18 7.27 1.78 4.07 6.57 6.95 0.90 8.10 5.25 8.98 2.20 2.83 7.18 0.69 9.68 7.63 5.30 6.82 3.52 3.18 9.38 8.61 3.26 8.80 0.17  
4.66 2.92 5.48 5.07 1.75 8.57 6.78 8.93 3.35 0.44 3.72 3.67 8.85 6.34 5.23 5.39 7.38 4.06 9.04 2.25 9.23 3.23 6.63 0.03 4.51

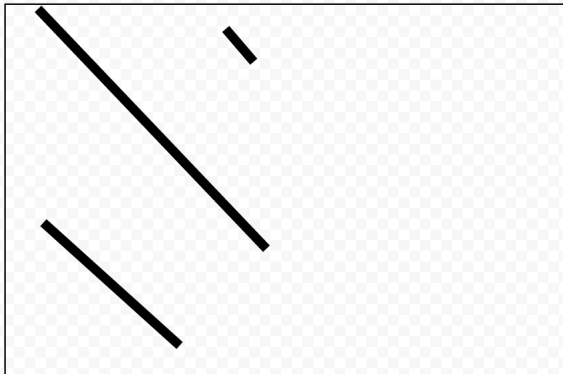
6.27 9.87 7.55 9.63 8.67  
7.72 2.00 9.83 0.60 3.84  
9.76 5.04 8.77 7.18 9.64  
5.59 4.32 1.35 5.91 2.34  
8.87 8.58 4.14 5.24 9.31  
8.52 0.74 5.75 1.69 5.16  
6.73 7.25 6.21 2.93 9.58  
6.70 5.33 9.32 3.69 4.22  
3.63 2.07 6.30 3.89 0.75  
9.33 9.80 8.53 5.18 8.12  
6.72 3.21 0.11 1.53 7.08  
6.79 8.49 5.09 6.68 2.78  
8.04 4.90 8.69 3.84 4.89  
8.62 6.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.62  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24  
7.10 6.70 2.97 6.58 1.21  
4.51 7.72 5.42 5.47 2.19  
5.25 9.57 9.30 0.97 4.49  
2.06 7.91 7.68 8.60 2.51

В случае выполнения  $\sigma < \epsilon$  в ячейку матрицы  
гомологии заносится true, иначе false.  
Для удобства графического представления  
будем закрашивать ячейку в черный цвет.

# Спектрально-аналитический метод

6.27 9.87 7.55 9.63 8.67  
7.72 2.00 9.83 0.60 3.84  
9.76 5.04 8.77 7.18 9.64  
5.59 4.32 1.35 5.91 2.34  
8.87 8.58 4.14 5.24 9.31  
8.52 0.74 5.75 1.69 5.16  
6.73 7.25 6.21 2.93 9.58  
6.70 5.33 9.32 3.69 4.22  
3.63 2.07 6.30 3.89 0.75  
9.33 9.80 8.53 5.18 8.12  
6.72 3.21 0.11 1.53 7.08  
6.79 8.49 5.09 6.68 2.78  
8.04 4.90 8.69 3.84 4.89  
8.62 6.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.62  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24  
7.10 6.70 2.97 6.58 1.21  
4.51 7.72 5.42 5.47 2.19  
5.25 9.57 9.30 0.97 4.49  
2.06 7.91 7.68 8.60 2.51

8.23 3.46 5.61 2.32 4.56 3.52 2.88 3.64 6.09 5.95 4.63 1.22 6.94 6.46 5.94 5.03 8.63 0.50 5.86 0.64 4.95 6.35 6.35 8.21 8.57  
7.62 4.12 0.60 8.66 5.56 1.97 6.15 4.49 7.22 4.70 7.10 3.58 2.42 1.70 4.22 2.97 9.96 9.21 4.22 0.75 3.87 2.53 5.32 8.26 3.13  
0.71 3.35 5.21 0.73 8.58 7.18 0.83 0.02 6.57 7.77 3.27 7.88 1.72 2.61 6.01 3.82 5.41 2.84 6.54 8.00 7.47 3.93 4.48 5.74 6.95  
9.18 7.27 1.78 4.07 6.57 6.95 0.90 8.10 5.25 8.98 2.20 2.83 7.18 0.69 9.68 7.63 5.30 6.82 3.52 3.18 9.38 8.61 3.26 8.80 0.17  
4.66 2.92 5.48 5.07 1.75 8.57 6.78 8.93 3.35 0.44 3.72 3.67 8.85 6.34 5.23 5.39 7.38 4.06 9.04 2.25 9.23 3.23 6.63 0.03 4.51



# Спектрально-аналитический метод

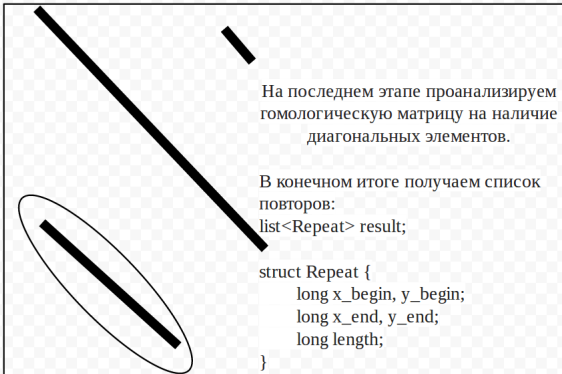
- 1 Получение GC/GA профиля биологической последовательности.
- 2 Построение спектров.
- 3 Сравнение спектров и построение гомологической матрицы.
- 4 Анализ гомологической матрицы.



# Спектрально-аналитический метод

```
8.23 3.46 5.61 2.32 4.56 3.52 2.88 3.64 6.09 5.95 4.63 1.22 6.94 6.46 5.94 5.03 8.63 0.50 5.86 0.64 4.95 6.35 6.35 8.21 8.57
7.62 4.12 0.60 8.66 5.56 1.97 6.15 4.49 7.22 4.70 7.10 3.58 2.42 1.70 4.22 2.97 9.96 9.21 4.22 0.75 3.87 2.53 5.32 8.26 3.13
0.71 3.35 5.21 0.73 8.58 7.18 0.83 0.02 6.57 7.77 3.27 7.88 1.72 2.61 6.01 3.82 5.41 2.84 6.54 8.00 7.47 3.93 4.48 5.74 6.95
9.18 7.27 1.78 4.07 6.57 6.95 0.90 8.10 5.25 8.98 2.20 2.83 7.18 0.69 9.68 7.63 5.30 6.82 3.52 3.18 9.38 8.61 3.26 8.80 0.17
4.66 2.92 5.48 5.07 1.75 8.57 6.78 8.93 3.35 0.44 3.72 3.67 8.85 6.34 5.23 5.39 7.38 4.06 9.04 2.25 9.23 3.23 6.63 0.03 4.51
```

```
6.27 9.87 7.55 9.63 8.67
7.72 2.00 9.83 0.60 3.84
9.76 5.04 8.77 7.18 9.64
5.59 4.32 1.35 5.91 2.34
8.87 8.58 4.14 5.24 9.31
8.52 0.74 5.75 1.69 5.16
6.73 7.25 6.21 2.93 9.58
6.70 5.33 9.32 3.69 4.22
3.63 2.07 6.30 3.89 0.75
9.33 9.80 8.53 5.18 8.12
6.72 3.21 0.11 1.53 7.08
6.79 8.49 5.09 6.68 2.78
8.04 4.90 8.69 3.84 4.89
8.62 6.79 3.67 4.36 0.69
0.96 2.16 6.33 4.25 6.23
0.43 9.14 2.11 8.77 1.69
7.91 5.28 2.14 1.59 4.16
7.94 9.75 4.58 7.35 9.09
5.85 7.86 4.87 2.81 5.91
7.30 4.81 8.52 1.53 3.46
2.38 0.82 1.51 8.22 4.22
2.45 6.85 5.05 0.27 5.35
2.36 10.00 5.23 4.12 8.62
2.46 8.98 3.14 8.67 9.35
8.68 2.95 2.10 7.24 5.87
8.55 5.24 5.40 7.07 7.24
7.10 6.70 2.97 6.58 1.21
4.51 7.72 5.42 5.47 2.19
5.25 9.57 9.30 0.97 4.49
2.06 7.91 7.68 8.60 2.51
```



# Схема параллельного алгоритма поиска повторов

На вход алгоритму подаются:

- Две биологические последовательности (массив символов четырехбуквенного алфавита A, T, G, C)
  - Как один из вариантов - одна последовательность, т.е. последовательность сравнивается сама с собой
- Параметры алгоритма
  - Длина скользящего окна профилировки
  - Длина скользящего окна построения спектров
  - Шаг сдвига окна построения спектров
  - $\varepsilon$  - точность сравнения спектров
  - Параметры фильтрации найденных повторов

# Схема параллельного алгоритма поиска повторов

```
char sequence[N]; // A, T, G, C
```

```
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGCGACGATA
```

# Схема параллельного алгоритма поиска повторов

```
char sequence[N]; // A, T, G, C
```

GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTCAGTCAGTCGAGACGATA

GTAGCATGGTAGAGTAGCA

TCGATCGATGCATGCACATGC

CGATGCATGCATCGCATGCTA

GCTACGTAGCTCAGTCGTA

GAGCTCAGTCGGAACGATA

Равномерно разбиваем входную биологическую последовательность на N равных частей.



# Схема параллельного алгоритма поиска повторов

char sequence[N]; // A, T, G, C

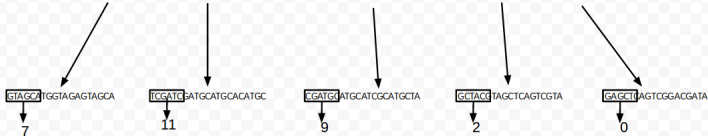


Асинхронная посылка окна соседу слева

# Схема параллельного алгоритма поиска повторов

char sequence[N]; // A, T, G, C

GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCTACGTAGCATCGATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTCAGTCAGTCGACGATA

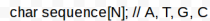


# Схема параллельного алгоритма поиска повторов

char sequence[N]; // A, T, G, C



## Схема параллельного алгоритма поиска повторов



GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACGTAGCATCGATGCATGCATCGTAGCATGCTAGCTACGTAGCTCAGTCGTAGCTCAGTCGGACGATA



### Асинхронная посылка окна соседу слева

# Схема параллельного алгоритма поиска повторов

char sequence[N]; // A, T, G, C

GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACTAGCATCGATGCATGCATGCTAGCATGCTACTAGCTAGCTCAGTCGTAAGCTCAGTCGGAACGATA

GTAGCATGGTAGAGATAGCA

5 6 7 8 9 10 11 10 9 8 9 8

0.27 9.87 7.55 9.63 8.67  
7.72 2.00 9.83 0.60 3.84  
9.76 5.04 8.77 7.18 9.64  
5.59 4.32 1.35 5.91 2.34  
8.87 8.58 4.14 5.24 9.31  
8.52 0.74 5.75 1.69 5.16  
6.73 7.25 6.21 2.93 9.58  
6.70 5.33 9.32 3.69 4.22  
3.62 2.07 6.30 3.89 0.75  
9.33 9.80 8.53 5.18 8.12  
6.72 3.21 0.11 1.53 7.08  
6.79 8.49 6.09 6.68 2.78  
8.04 4.90 8.69 3.84 4.89  
8.62 0.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09

TCGATCGATGCATGCACATGC

8 9 10 11 12 3 4 3 2 3

8.62 6.70 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

CGATGCATGCATGCATGCTA

7 8 9 10 11 12 11 10 9

8.62 6.70 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

GCTAGCTAGCTCAGTCGTA

4 5 6 7 8 9 10 11

8.62 6.70 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

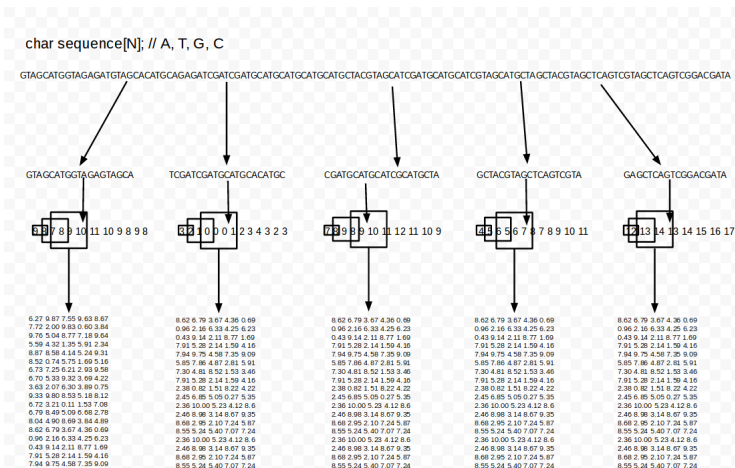
GAGCTCAGTCGGAACGATA

1 2 13 14 15 16 17

8.62 6.70 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.69  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

На каждом процессе имеем свой набор спектров

# Схема параллельного алгоритма поиска повторов



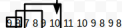
Такие же этапы вычислений происходят и со второй последовательностью

# Схема параллельного алгоритма поиска повторов

char sequence[N]; // A, T, G, C

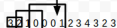
GTAGCATGGTAGAGATGTAGCACATGCAGAGATCGATCGATGCATGCATGCATGCATGCTACTAGCATCGATGCATGCATGCTAGCATGCTACTAGCTCAGTCGTCAGTCAGTCGGACGATA

GTAGCATGGTAGAGTAGCA



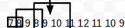
6.27 9.87 7.55 9.63 8.67  
7.72 2.00 9.83 0.60 3.84  
9.70 5.04 8.77 7.18 9.64  
5.59 4.32 1.35 5.91 2.34  
8.87 8.58 4.14 5.24 9.31  
8.52 0.74 5.75 1.09 5.10  
6.73 7.25 6.21 2.93 9.59  
6.70 5.33 9.32 3.69 4.22  
3.63 2.07 6.30 3.89 0.75  
9.33 9.80 8.53 5.18 8.12  
6.72 3.21 0.11 1.53 7.08  
6.79 8.45 5.09 6.98 2.78  
8.04 4.90 8.69 3.84 4.89  
8.62 0.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.09  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
8.55 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

TCGATCGATGCATGCACATGC



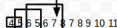
8.62 6.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.09  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

CGATGCATGCATGCATGCTA



8.62 6.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.09  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

GCTACTAGCTCAGTCGTA



8.62 6.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.09  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

GAGTCAGTCGGACGATA



8.62 6.79 3.67 4.36 0.69  
0.96 2.16 6.33 4.25 6.23  
0.43 9.14 2.11 8.77 1.09  
7.91 5.28 2.14 1.59 4.16  
7.94 9.75 4.58 7.35 9.09  
5.85 7.86 4.87 2.81 5.91  
7.30 4.81 8.52 1.53 3.46  
7.91 5.28 2.14 1.59 4.16  
2.38 0.82 1.51 8.22 4.22  
2.45 6.85 5.05 0.27 5.35  
2.36 10.00 5.23 4.12 8.6  
2.46 8.98 3.14 8.67 9.35  
8.68 2.95 2.10 7.24 5.87  
8.55 5.24 5.40 7.07 7.24

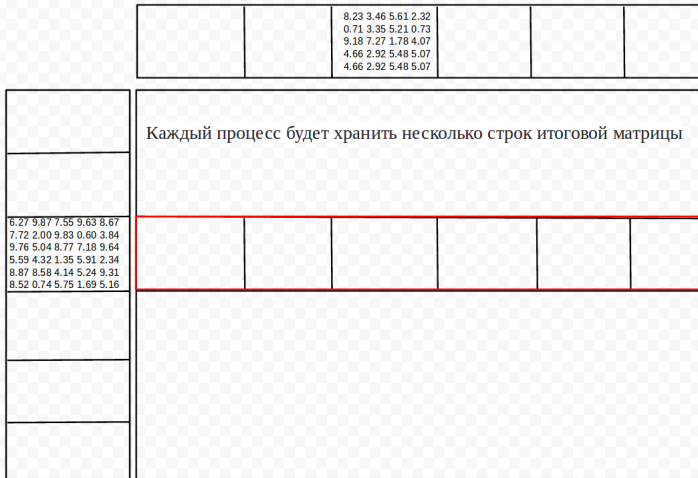
Таким образом на каждом процессе имеем два набор спектров

## Схема параллельного алгоритма поиска повторов

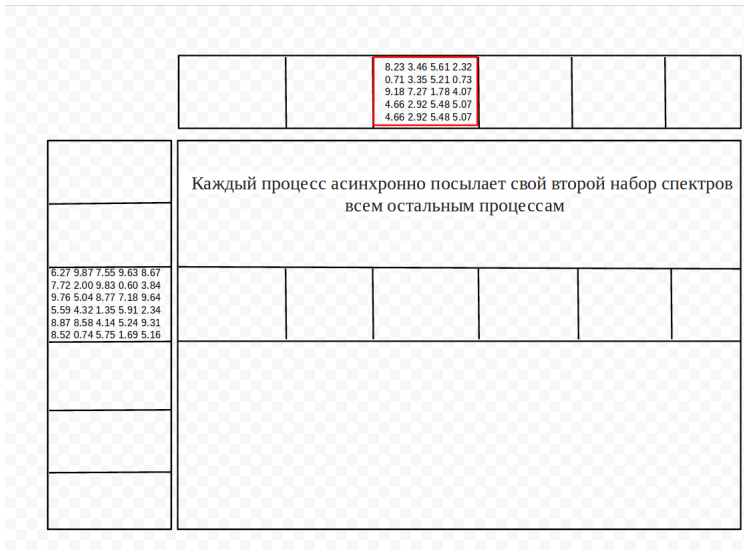
		8.23 3.46 5.61 2.32 0.71 3.35 5.21 0.73 9.18 7.27 1.78 4.07 4.66 2.92 5.48 5.07 4.66 2.92 5.48 5.07			
6.27 9.87 7.55 9.63 8.67 7.72 2.00 9.83 0.60 3.84 9.76 5.04 8.77 7.18 9.64 5.59 4.32 1.35 5.91 2.34 8.87 8.58 4.14 5.24 9.31 8.52 0.74 5.75 1.69 5.16					



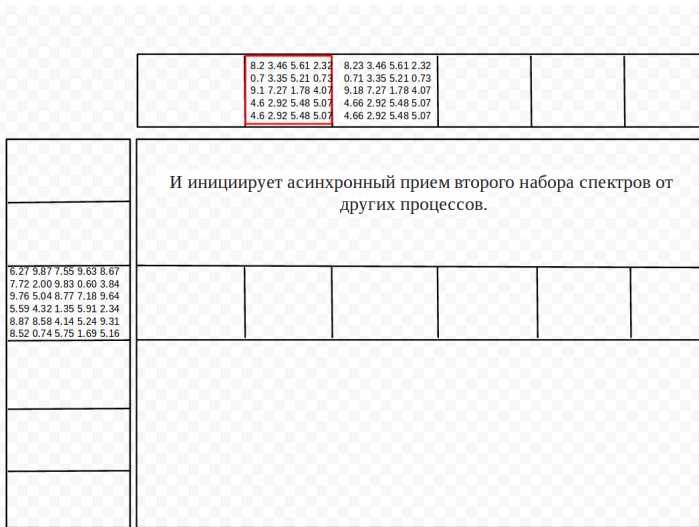
# Схема параллельного алгоритма поиска повторов



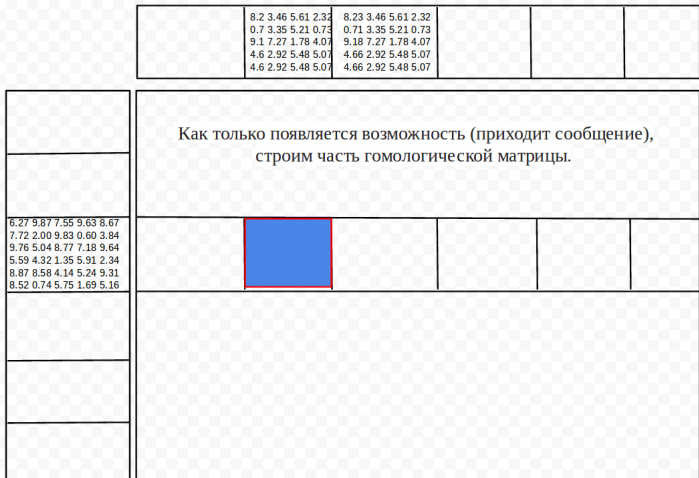
# Схема параллельного алгоритма поиска повторов



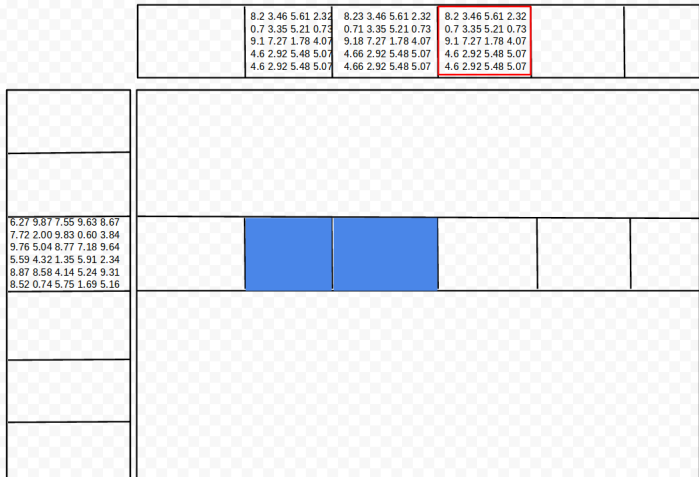
# Схема параллельного алгоритма поиска повторов



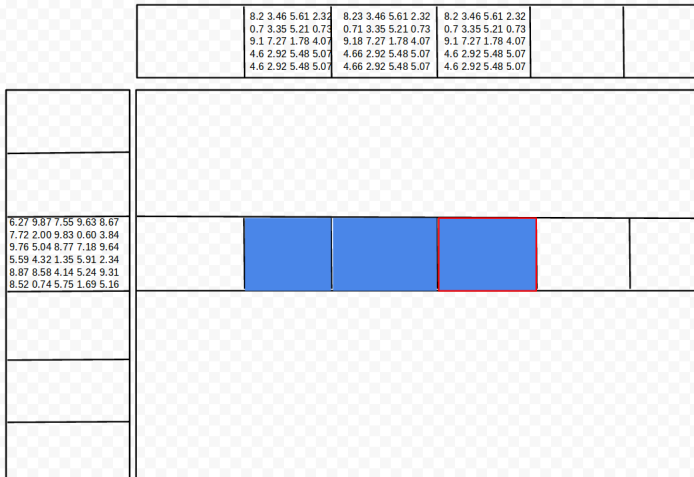
# Схема параллельного алгоритма поиска повторов



# Схема параллельного алгоритма поиска повторов



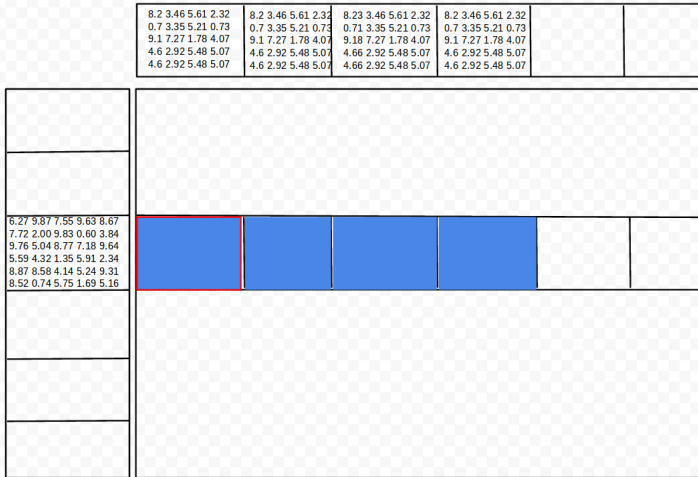
# Схема параллельного алгоритма поиска повторов



# Схема параллельного алгоритма поиска повторов

	8.2 3.46 5.61 2.32 0.7 3.35 5.21 0.73 9.1 7.27 1.78 4.07 4.6 2.92 5.48 5.07 4.6 2.92 5.48 5.07	8.2 3.46 5.61 2.32 0.7 3.35 5.21 0.73 9.1 7.27 1.78 4.07 4.6 2.92 5.48 5.07 4.6 2.92 5.48 5.07	8.23 3.46 5.61 2.32 0.71 3.35 5.21 0.73 9.18 7.27 1.78 4.07 4.66 2.92 5.48 5.07 4.66 2.92 5.48 5.07	8.2 3.46 5.61 2.32 0.7 3.35 5.21 0.73 9.1 7.27 1.78 4.07 4.6 2.92 5.48 5.07 4.6 2.92 5.48 5.07		
6.27 9.87 7.55 9.63 8.67 7.72 2.00 9.83 0.60 3.84 9.76 5.04 8.77 7.18 9.64 5.59 4.32 1.35 5.91 2.34 8.87 8.58 4.14 5.24 9.31 8.52 0.74 5.75 1.69 5.16						

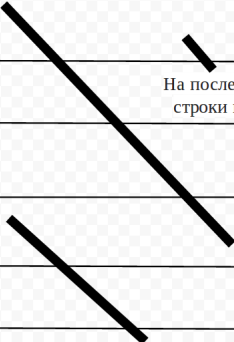
# Схема параллельного алгоритма поиска повторов



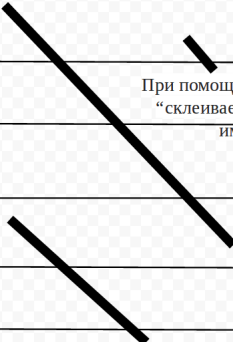


# Схема параллельного алгоритма поиска повторов

8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.23 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.6 2.3
0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.71 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.1 0.7
9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.18 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.8 4.0
4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.66 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.9 4.48 5.0
4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.66 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.2 5.8 5.07

	 <p>На последнем этапе анализируем имеющиеся строки гомологической матрицы на наличие диагональных элементов и получаем список повторов.</p>
6.27 9.87 7.55 9.63 8.67 7.72 2.00 9.83 0.60 3.84 9.76 5.04 8.77 7.18 9.64 5.59 4.32 1.35 5.91 2.34 8.87 8.58 4.14 5.24 9.31 8.52 0.74 5.75 1.69 5.16	

# Схема параллельного алгоритма поиска повторов

	<table><tr><td>8.2 3.46 5.61 2.32</td><td>8.2 3.46 5.61 2.32</td><td>8.23 3.46 5.61 2.32</td><td>8.2 3.46 5.61 2.32</td><td>8.2 3.46 5.61 2.32</td><td>8.2 3.46 5.61 2.32</td><td>8.2 3.46 5.61 2.32</td><td>8.2 3.46 5.61 2.32</td><td>8.2 3.46 5.61 2.32</td></tr><tr><td>0.7 3.35 5.21 0.73</td><td>0.7 3.35 5.21 0.73</td><td>0.71 3.35 5.21 0.73</td><td>0.7 3.35 5.21 0.73</td><td>0.7 3.35 5.21 0.73</td><td>0.7 3.35 5.21 0.73</td><td>0.7 3.35 5.21 0.73</td><td>0.7 3.35 5.21 0.73</td><td>0.7 3.35 5.1 0.7</td></tr><tr><td>9.1 7.27 1.78 4.07</td><td>9.1 7.27 1.78 4.07</td><td>9.18 7.27 1.78 4.07</td><td>9.1 7.27 1.78 4.07</td><td>9.1 7.27 1.78 4.07</td><td>9.1 7.27 1.78 4.07</td><td>9.1 7.27 1.78 4.07</td><td>9.1 7.27 1.78 4.07</td><td>9.1 7.27 1.8 4.0</td></tr><tr><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.66 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.9 4.48 5.0</td></tr><tr><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.66 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.92 5.48 5.07</td><td>4.6 2.2 5.8 5.07</td></tr></table>	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.23 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.71 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.1 0.7	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.18 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.8 4.0	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.66 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.9 4.48 5.0	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.66 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.2 5.8 5.07
8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.23 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32	8.2 3.46 5.61 2.32																																						
0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.71 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.21 0.73	0.7 3.35 5.1 0.7																																						
9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.18 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.78 4.07	9.1 7.27 1.8 4.0																																						
4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.66 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.9 4.48 5.0																																						
4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.66 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.92 5.48 5.07	4.6 2.2 5.8 5.07																																						
	 <p>При помощи односторонних коммуникаций MPI “склеиваем” повторы, т.е. объединяем повторы имеющие разрывы на стыке процессов в единые повторы.</p>																																													
6.27 9.87 7.55 9.63 8.67 7.72 2.00 9.83 0.60 3.84 9.76 5.04 8.77 7.18 9.64 5.59 4.32 1.35 5.91 2.34 8.87 8.58 4.14 5.24 9.31 8.52 0.74 5.75 1.69 5.16																																														

# Структура программной реализации параллельного алгоритма

Параллельная реализация программы выполнена на C++ с использованием технологии параллельного программирования MPI и технологии CUDA-C для использования графических ускорителей Nvidia.

Объем кода всего проекта (основная программа + GUI) составляет 6 тыс. строк.

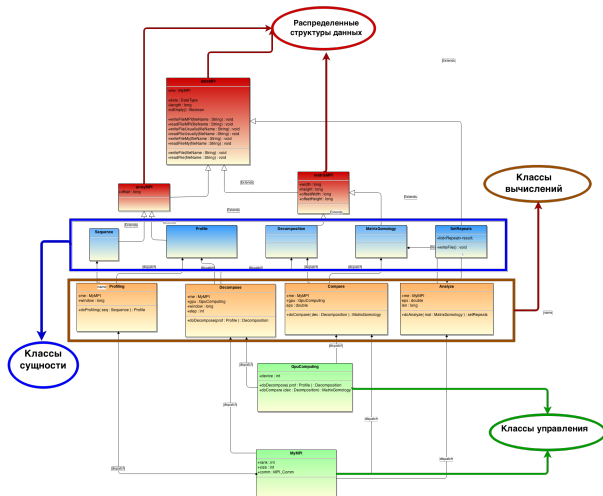
# Структура программной реализации параллельного алгоритма

В структуре классов можно выделить следующие группы:

- Шаблонные классы, описывающие распределенные структуры данных.
- Классы-сущности, специализирующие шаблонные классы.
- Классы-вычислители, преобразующие одни классы-сущности в другие классы-сущности.
- Управляющие классы, через которые осуществляется взаимодействие с MPI и GPU устройством.

Структура алгоритма позволяет выделить каждый этап в отдельный модуль. Таким образом работу программы можно начать с любого этапа и закончить на любом этапе.

# Структура программной реализации параллельного алгоритма



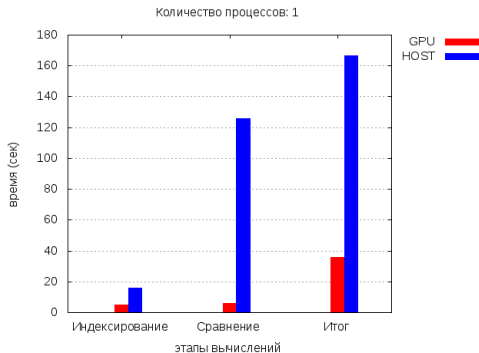
# Структура программной реализации параллельного алгоритма

Выделение в отдельные классы взаимодействия с MPI и GPU устройством позволило при компиляции отключить возможности эти классов и иметь следующие варианты сборки:

- последовательная программа
- последовательная программа, использующая графические процессоры
- параллельная программа без использования графических процессоров
- параллельная программа с использованием графических процессоров

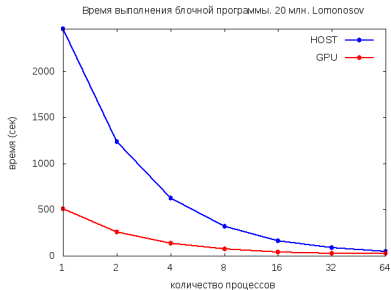
Данная возможность позволяет существенно расширить список вычислительных платформ на которых может работать программа.

# Результаты тестирования программной реализации

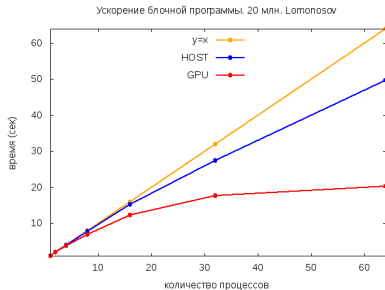


Сравнение HOST-программы и GPU-программы

# Результаты тестирования программной реализации



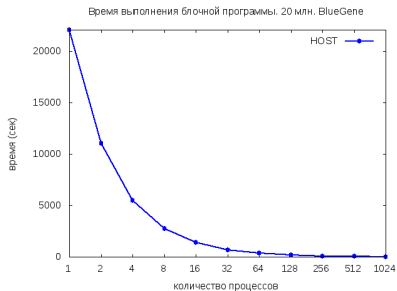
Сильная масштабируемость GPU- и HOST-программы



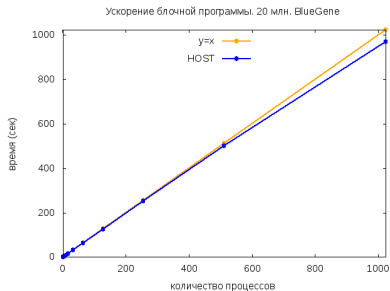
Ускорение GPU- и HOST-программы



# Результаты тестирования программной реализации

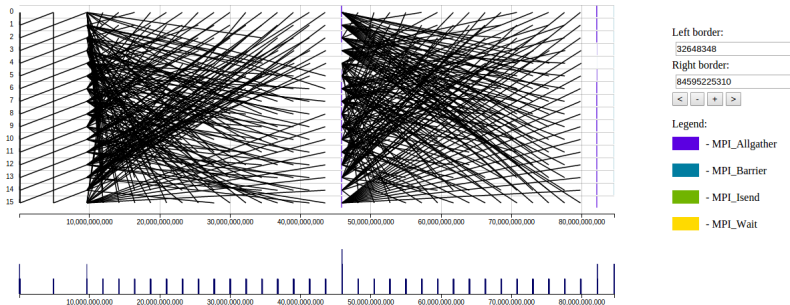


Сильная масштабируемость  
HOST-программы



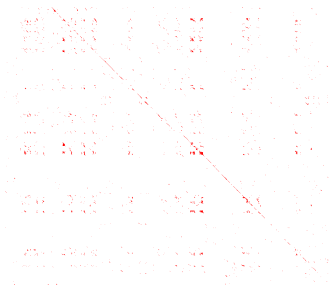
Ускорение HOST-программы

# Результаты тестирования программной реализации

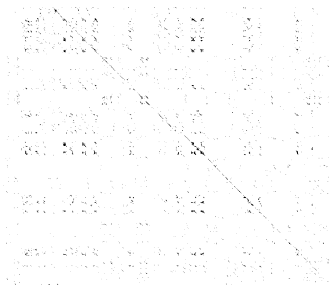


Трасса программы

# Полученные результаты



(a)



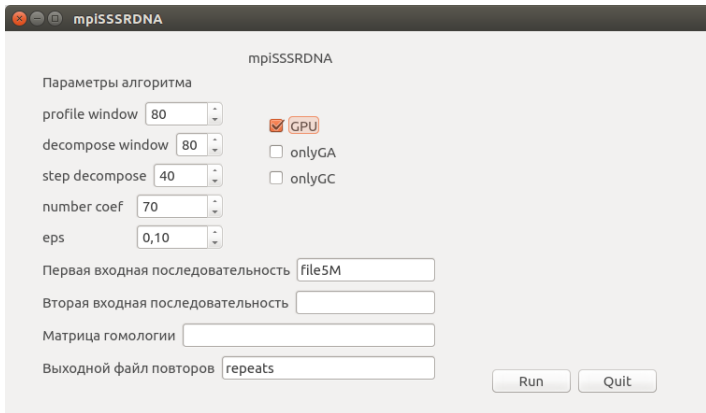
(b)

Гомологическая матрица сравнения *Mus musculus* и *Rattus norvegicus*. Длина последовательностей 172.5 Мб и 143.6 Мб соответственно. Параметры алгоритма:

$W1 : 300000$ ,  $d1 : 64$ ,  $W2 : 800000$ ,  $d2 : 160000$ ,  $coef : 5$ ,  $\varepsilon 4 * 10^{-5}$ ; (a) последовательная программа; (b) параллельная программа;

# Графический интерфейс

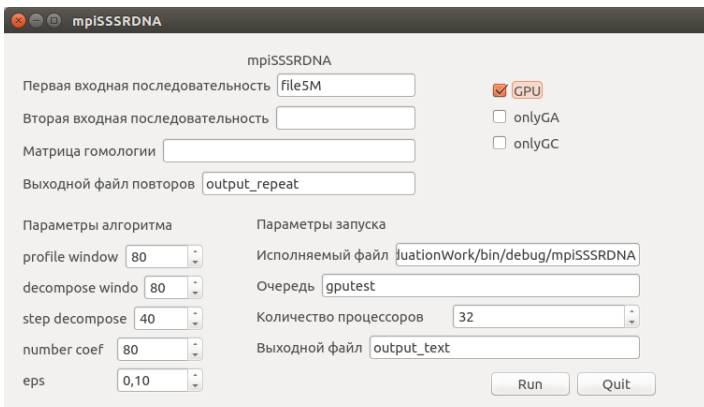
Графический интерфейс выполнен с использованием кроссплатформенной библиотеки QtGUI.



# Графический интерфейс

Так же был разработан графический интерфейс использующий возможности библиотеки libssh2.

Данный интерфейс позволяет пользователю запускать графический интерфейс на локальной машине и отсылать команду на исполнение на удаленный сервер.



The screenshot shows a window titled "mpiSSSRDNA" with a dark header bar. The interface is divided into several sections for configuring the application:

- mpiSSSRDNA** (Section Header):
  - Первая входная последовательность:
  - Вторая входная последовательность:
  - Матрица гомологии:
  - Выходной файл повторов:
  - GPU: ☒ GPU, ☐ onlyGA, ☐ onlyGC
- Параметры алгоритма** (Algorithm Parameters):
  - profile window:
  - decompose windo:
  - step decompose:
  - number coef:
  - eps:
- Параметры запуска** (Execution Parameters):
  - Исполняемый файл:
  - Очередь:
  - Количество процессоров:
  - Выходной файл:

At the bottom right, there are two buttons: "Run" and "Quit".

# Основные результаты

- Разработан и реализован на суперкомпьютерах Ломоносов, BlueGene/P и многопроцессорной рабочей станции параллельный алгоритм поиска повторов в биологических последовательностях. Алгоритм реализует парные сравнения последовательностей большого размера. Параллельная реализация алгоритма выполнена с использованием технологии MPI с возможностью подключения cuda-модулей для использования графических ускорителей.
- Проведено тестирование разработанного алгоритма на искусственных данных, а так же на решении реальной задачи сравнения биологических последовательностей крысы и мыши. Полученные результаты параллельного метода совпадают с последовательной реализацией метода.

# Основные результаты

- Проведено исследование эффективности и масштабируемости разработанной параллельной программы.
- Показано, что использование графических процессоров позволяет ускорить алгоритм в 5 раз.
- Разработан графический интерфейс для параллельной программы, позволяющий пользователю запускать программу как на локальном, так и на удаленной системе. Интерфейс реализован с использованием кроссплатформенной библиотекой Qt, что позволяет использовать программу в различных операционных средах.

# Основные результаты

- Часть работы с использованием графических процессоров была поддержана компанией Nvidia в конкурсе студенческих работ и опубликована в альманахе <sup>3</sup>

---

<sup>3</sup>Cuda альманах. — <http://www.nvidia.ru/docs/IO/141194/CUDA—2015>.  
— февраль.



Спасибо за внимание!

