

# Полулокальное выравнивание с использованием внутрипроцессорного параллелизма

Дмитрий Орехов

Институт биоинформатики

Научный руководитель: Александр Тискин, The University of Warwick

25.05.2019

# Задача полулокального выравнивания

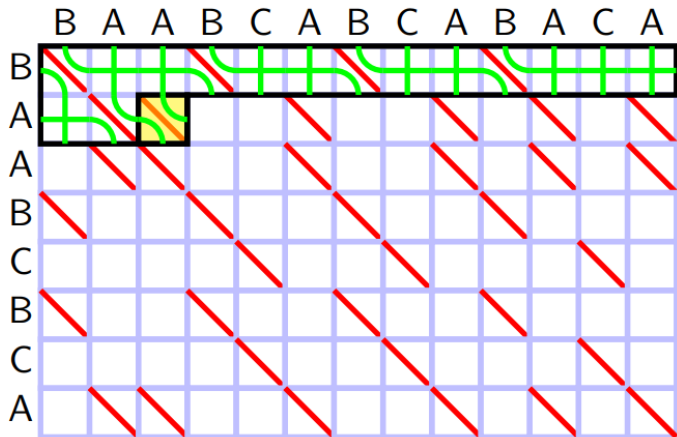
Получение выравниваний *Pattern* на *Text* для всех возможных окон фиксированной ширины  $w$ .

Где может использоваться:

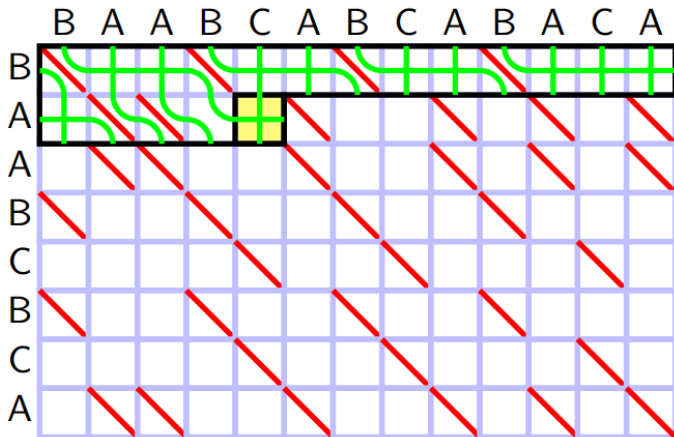
- Поиск ДНК или аминокислотных последовательностей.
- Выравнивание ридов на геном.

# Алгоритм Seaweed

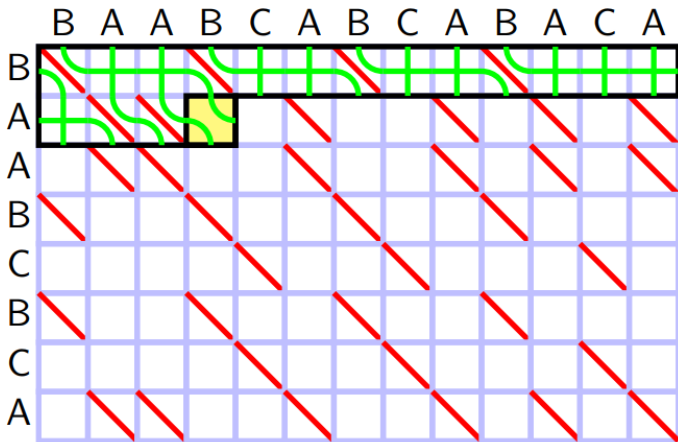
Match:



Mismatch:



Ранее пересекались:



Score выравнивания на окне ширины  $w$  равен разнице  $w$  и числа водорослей, начавшихся в данном окне и не вышедших за его пределы.

Алгоритм Seaweed поддерживает рациональные веса с небольшим знаменателем.

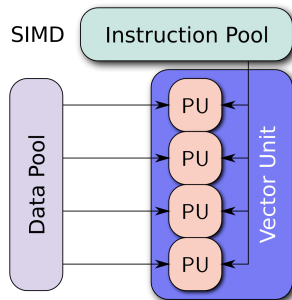
- Позволяет решать задачу выравнивания в парадигме Divide-and-conquer.
- Широкие возможности для параллелизма.
- Результат алгоритма - точное выравнивание.
- За алгоритмом стоит глубокая алгебраическая теория - разработка в этом направлении может позволить узнать что-то новое об алгоритмах на строках в целом.

## Single instruction multiple data (SIMD):

параллельное применение одной и той же операции к набору данных.

## Advanced Vector Instruction (AVX):

система команд для процессоров Intel и AMD, intrinsics функции в C и других языках - удобный способ для работы с ними.

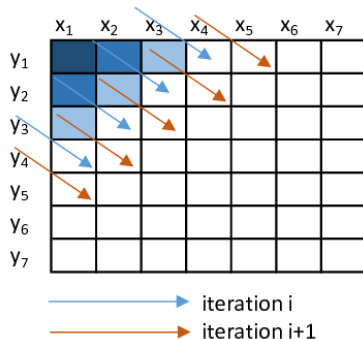




# Регистр 512 бит

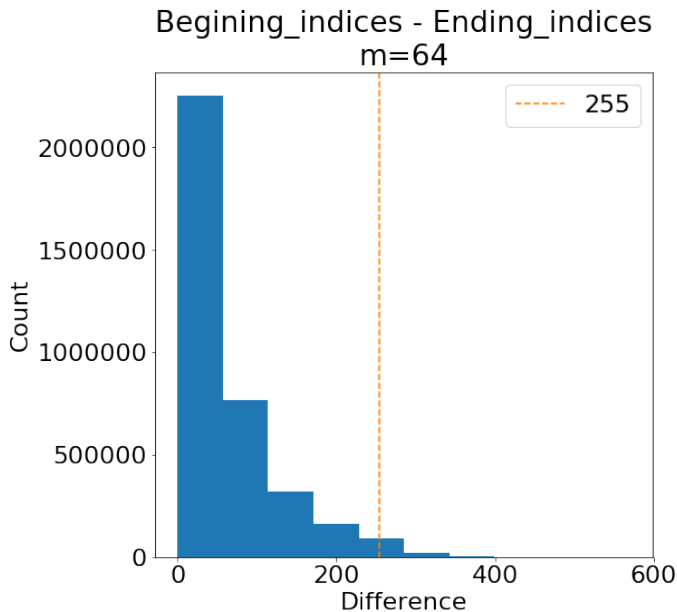
Современные процессоры с регистром 512 бит позволяют параллельно считать антидиагонали в матрице выравнивания для строк длиной до 64 символов.

Работа над проектом ведется с использованием Intel Software Development Emulator.



- Был реализован прототип Seaweed на Python3 с векторизацией в numpy.
- Реализован Seaweed на C с использованием AVX-512, данная реализация работает с короткими последовательностями из стандартного ввода.

# Проблема неразличимых водорослей



- Работает с файлами .fasta и .fastq
- Возможность выравнивать последовательности любой длины
- Time complexity:  $O(\frac{mN}{32})$
- Space complexity:  $O(N)$

- Приложение для командной строки на Python3 считывает аргументы и делает необходимый пэддинг
- В памяти хранится текст и два массива длины  $N$ , хранящие начала и концы водорослей
- Минимизируется число обращений к памяти, считывание новых данных происходит каждые 32 антидиагонали

# Проверка алгоритма

```
>0 <unknown description>  
POZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNO
```

```
Computing leaves  
Computing score  
Sorting score  
The pattern is: #####ONSARKERP_OTE_EHZ_KAK_TEATOBAR_MTIROGLA_UJALVARDZO  
Window_size = 64  
GGATACATGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNO: 20  
GATACATGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNO: 20  
ATACATGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOC: 20  
TACATGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOCG: 20  
ACATGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOCGG: 20  
CATGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOCGGG: 20  
ATGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOCGGGC: 20  
TGTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOCGGGCC: 20  
GTGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOCGGGCCC: 20  
TGTTTCPOZDRAVLAJU_ALGORITM_RABOTAET_KAK_ZHE_ETO_PREKRASNOCGGGCCCA: 20
```

# Спасибо за внимание!

GitHub: [https://github.com/DimaOrekhov/Seaweed\\_AVX512](https://github.com/DimaOrekhov/Seaweed_AVX512)

- 1 A. Tiskin, Semi-local longest common subsequences: A superglue for string comparison, URL:[https://www.dcs.warwick.ac.uk/~tiskin/pub/talks/semi\\_talk.pdf](https://www.dcs.warwick.ac.uk/~tiskin/pub/talks/semi_talk.pdf)
- 2 Ozsoy A., Chauhan A., Swamy M. Achieving teraCUPS on longest common subsequence problem using GPGPUs //2013 International Conference on Parallel and Distributed Systems. – IEEE, 2013. – С. 69-77.
- 3 Wikipedia contributors. (2019, March 18). SIMD. In Wikipedia, The Free Encyclopedia. Retrieved 20:57, April 4, 2019, from <https://en.wikipedia.org/w/index.php?title=SIMD&oldid=888335694>