Обзор алайнеров

Дмитрий Яковлев

EPAM Systems

19 октября 2016 г.

План

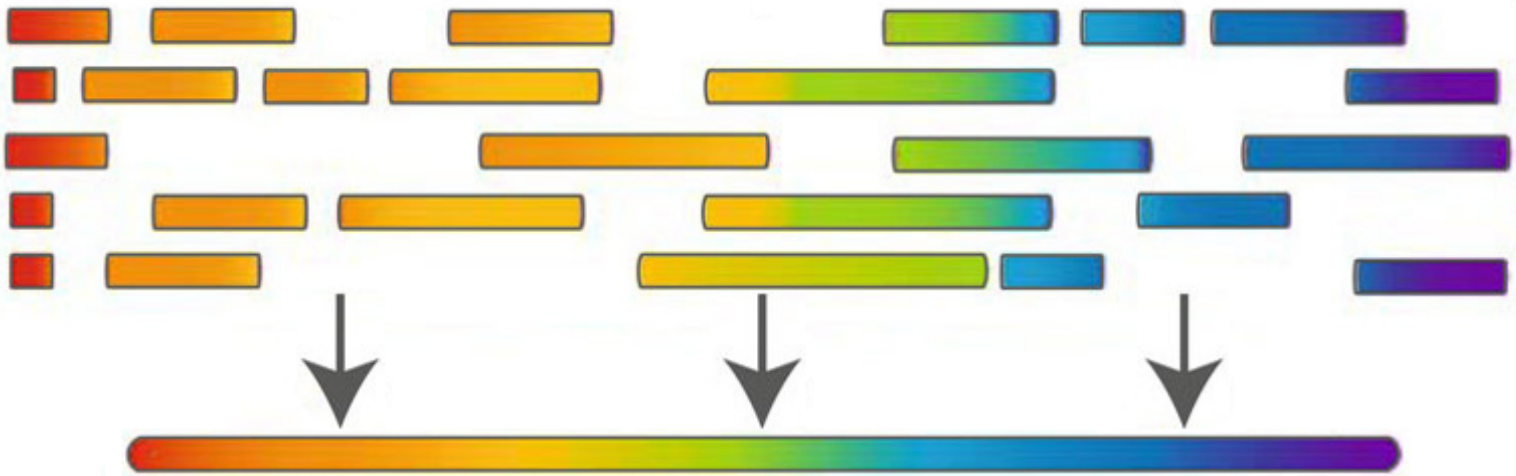
Введение

Цель

Найти наилучший для ассемблирования алайнер для датасета от BioNano.

Алайнер:

Ассемблер:



Разновидности моделей ошибок

Разновидности моделей ошибок

Виды ошибок:

- **Ошибка в ориентации карты** - распределение бернулли с параметром 0.5
- **Длина фрагмента**
  Пусть r фрагмент на референсе, тогда длина фрагмента $o \sim N(r, r\,\sigma^2)$
- **Лишние разрезы**
  Количество пропущенных разрезов - Пуассоновское распределение, расположение - равномерно по карте
- **Пропущенные разрезы** - распределение Бернулли
- **Пропущенные фрагменты** - распределение Бернулли

Модель ошибки на данных BioNano

Модель ошибок: общие сведения

- Группой ученых было рассмотрено 3 датасета карт от BioNano [2]
- С помощью RefAligner был построен референс
- Далее был проведён анализ модели ошибок

Модель ошибок: ошибка в длине фрагмента

Валуев:

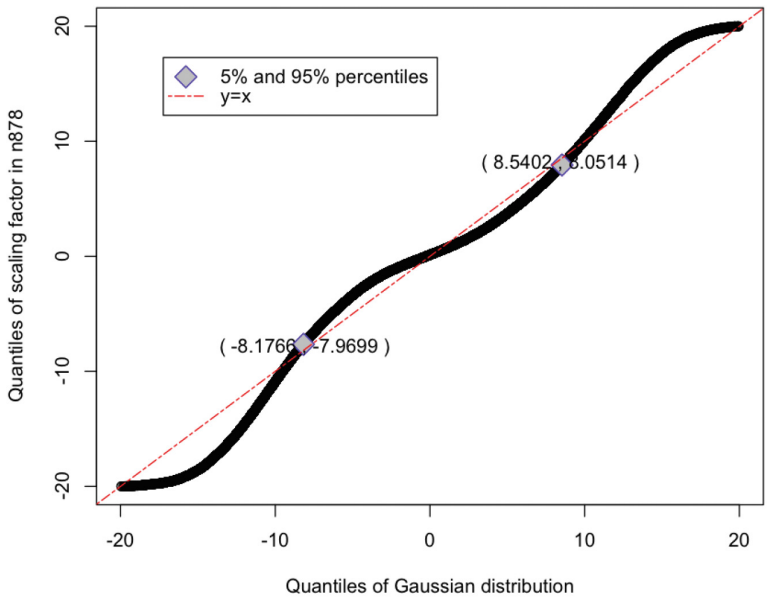
$$e_k = \frac{o_k - r_k}{\sqrt{r_k}} \sim N(0, \sigma)$$

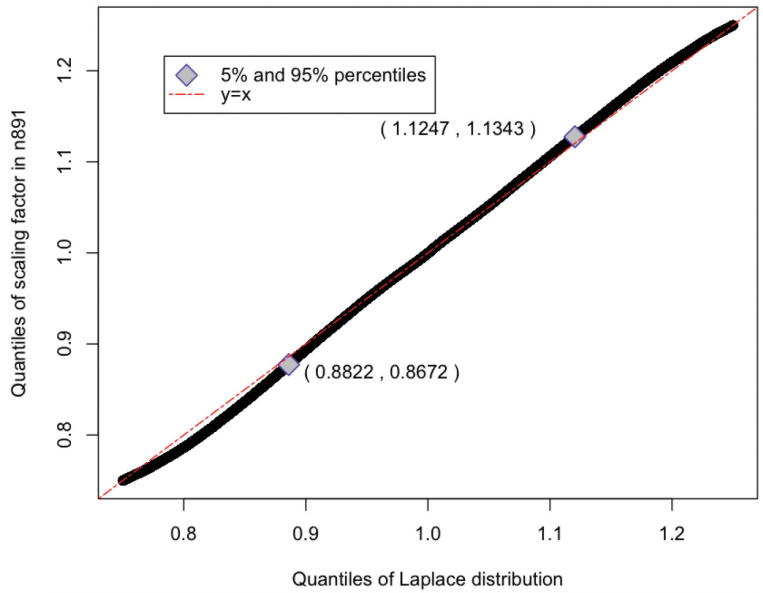
$$o_k \sim N(r_k, \sigma^2 r_k)$$

Новый подход:

$$s_k = \frac{o_k}{r_k}$$

$$s_k \sim Laplace(\mu, \beta)$$

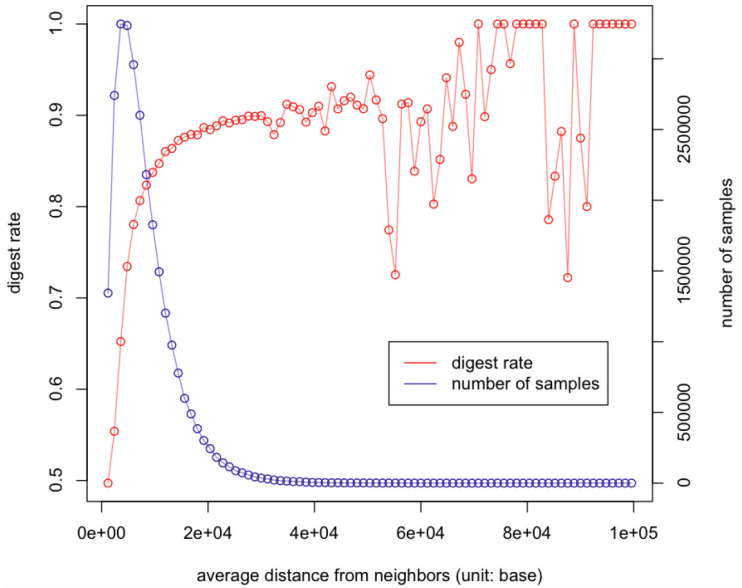


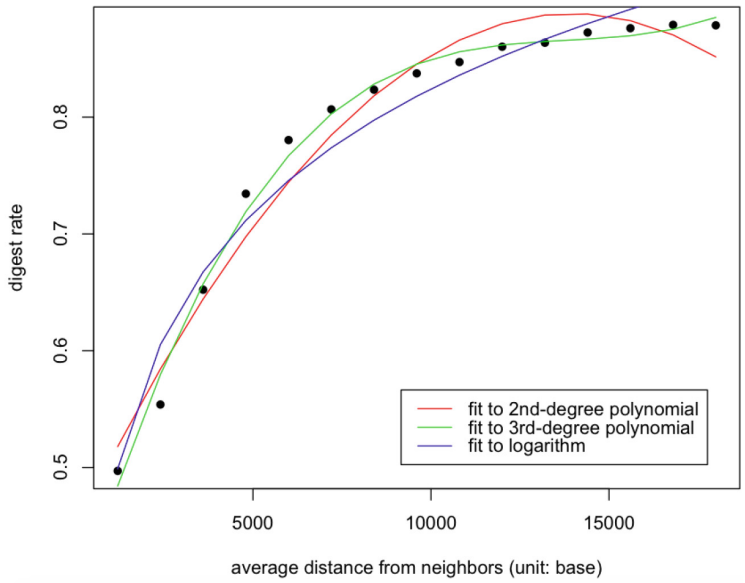


o_k и r_k - длины фрагментов на карте и референсе

Модель ошибок: пропущенные разрезы

Было замечено, что вероятность пропущенного разреза зависит от длины до соседних разрезов.

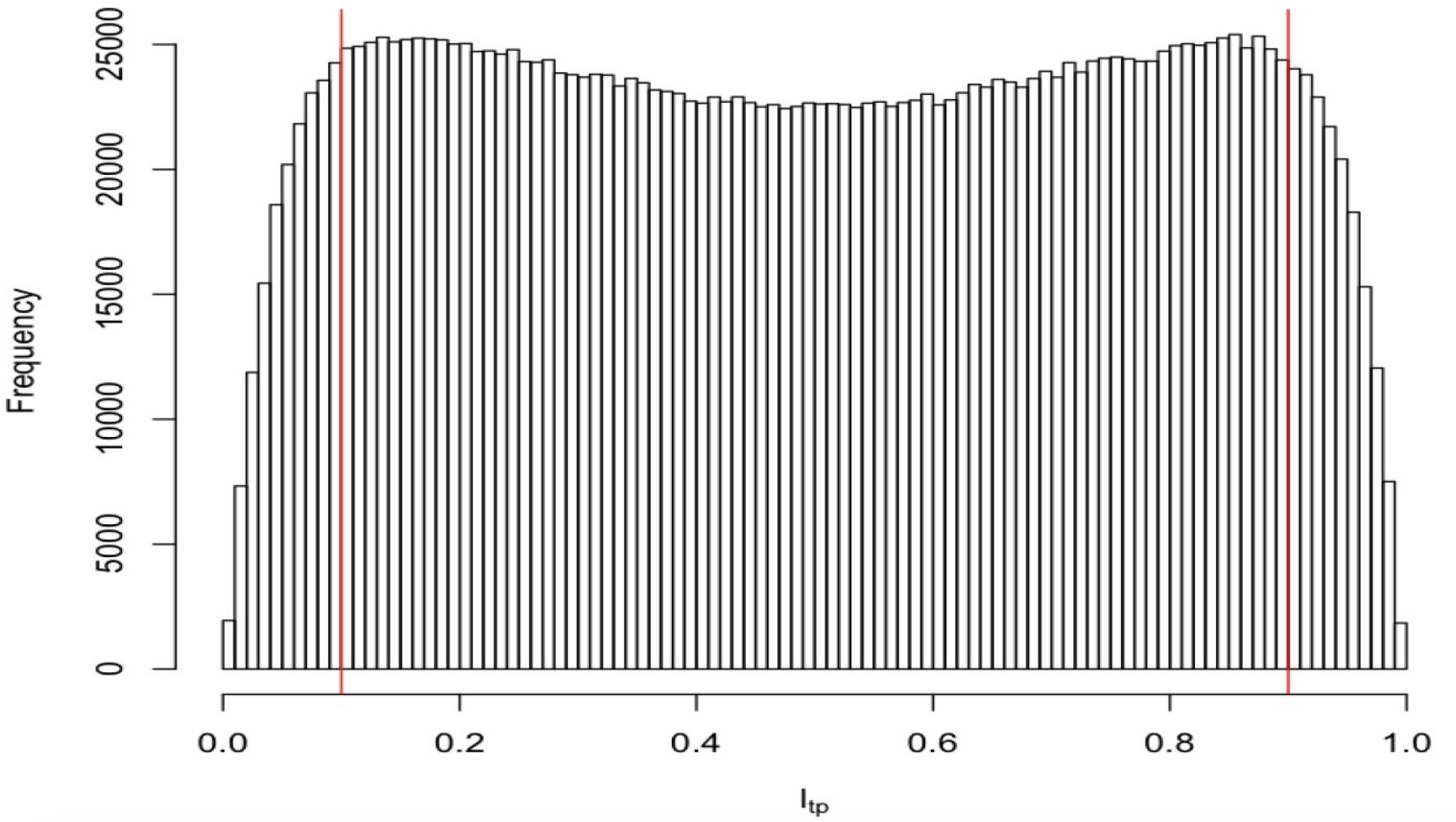




$$p_c(d_{avg}) = \alpha_3\, d_{avg}^3 + \alpha_2\, d_{avg}^2 + \alpha_1\, d_{avg} + \alpha_0$$

$$d_{avg} = \frac{\text{среднее расстояние до соседей}}{1200}$$

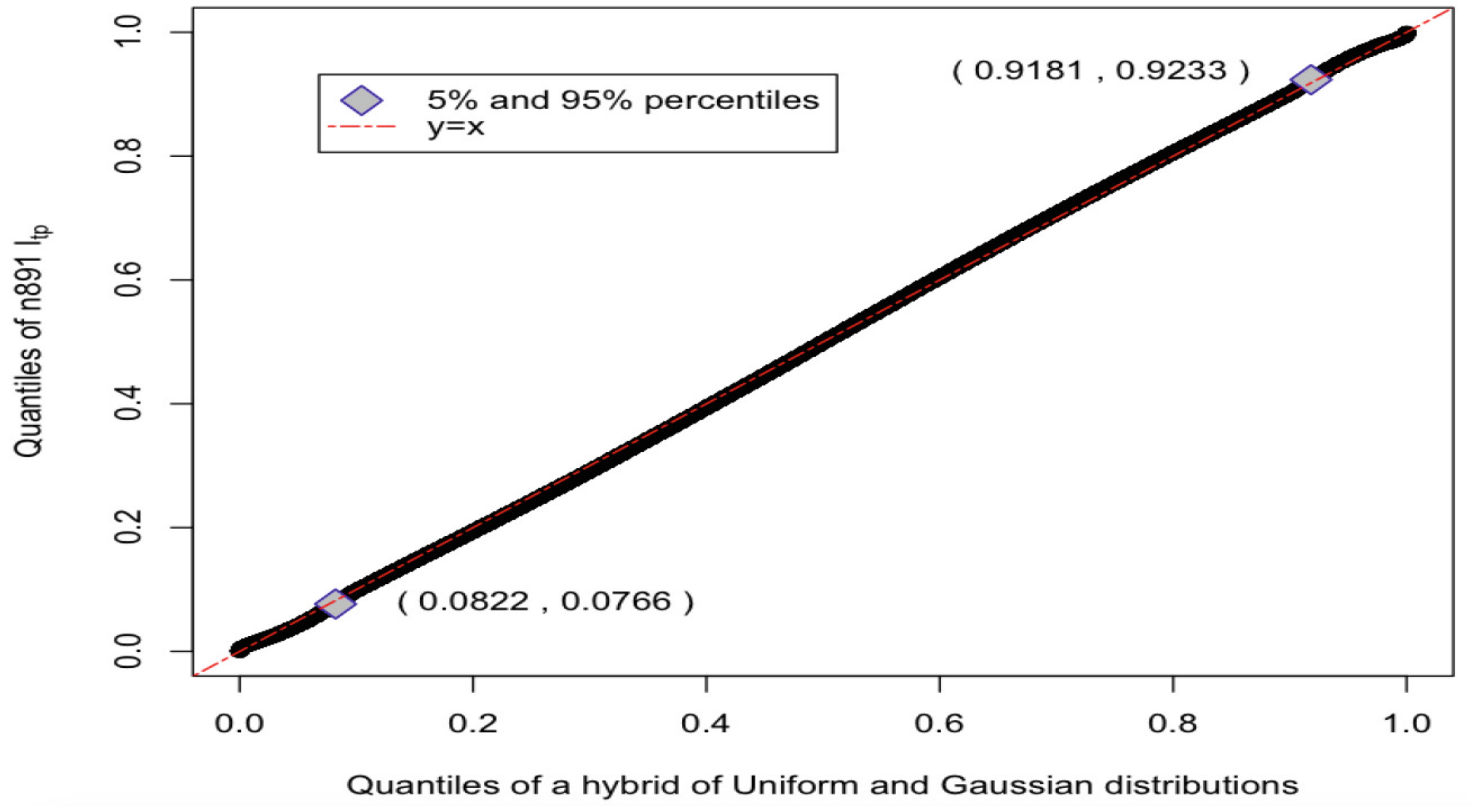
Модель ошибок: лишние разрезы (1)



$$l_{fp} = \frac{\text{расстояние от лишнего разреза до конца карты}}{\text{длина оптической карты}}$$

$$n_{fp} \sim 0.18\, Poisson(0) + 0.6\, Poisson(1) + 0.22\, Poisson(3)$$

Модель ошибок: лишние разрезы (2)



$$l_{\text{fp}} \sim \begin{cases} U[0.1, 0.9], & 0.1 \le l_{\text{fp}} \le 0.9, \text{ w.p. } 0.8852 \\ N(0.1, 0.044186), & l_{\text{fp}} < 0.1, \text{ w.p. } 0.0574 \\ N(0.9, 0.044186), & l_{\text{fp}} > 0.9, \text{ w.p. } 0.0574 \end{cases}$$

Алайнеры

Общие сведения

Классификация алайнеров:

- Использование матрицы выравнивания (алгоритм Смита-Ватермана)
- По наличию хешинга
- Модель ошибки

Общая сехама:

1. Определение мест на референсе, куда карта может быть выравнена (этап хешинга)
2. Построение выравнивания с использованием матрицы выравнивания
3. Определение значимости выравнивания
4. Выбор лучшего выравнивания

Общие сведения: ранее известные алайнеры

- **SOMA**
  Аналогичен OPTIMA
- **Gentig**
- **Valouev**
  На основе максимизации функции макссимального правдоподобия

OPTIMA

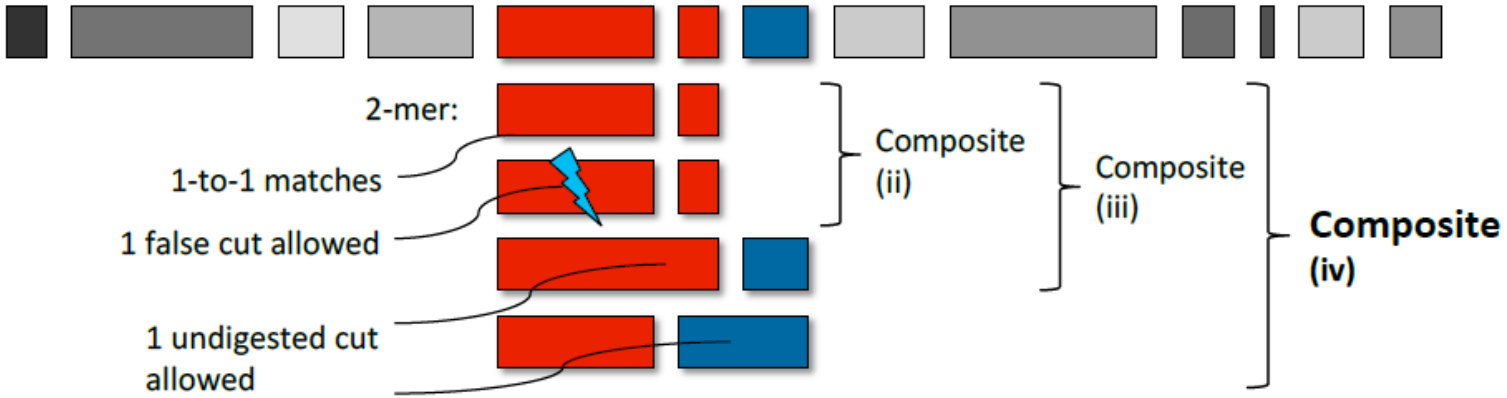
OPTIMA: Алгоритм

Этапы выравнивания:

- Поиск стартовых мест (сидов) для начала выравнивания
- Парное выравнивание карты с референсом
- Определение значимых выравниваний
- Объединение пересекающихся выравниваний

OPTIMA: Композитные сиды

Композитные сиды:



Определение.

Множество фрагментов o_k, o_{k+1}, ..., o_s **возможно совпадает** с множеством фрагментов r_l, r_{l+1}, ..., r_t :

$$\frac{\left| \sum_{i=k}^{s} o_i - \sum_{j=l}^{t} r_j \right|}{\sqrt{\sum_{j=l}^{t} \sigma_j^2}} \leq C_\sigma \qquad (1)$$

где C_σ - порог совпадения, σ_j - стандартное отклонение r_j

OPTIMA: Поиск стартовых сидов

Алгоритм поиска сидов для выравнивания:

- По референсу строятся композитные сиды и сортируются по первому элементу
- У карты берётся сид, по которому будем искать множество подходящих локаций (1) в референсе
- Бинарным поиском (по первому элементу) ищем множество подходящих сидов в референсе
- Далее линейно проверяем и оставляем только те, которые удовлетворяют (1)
- Таким образом получаем множество сидов на референсе, где карта может быть выравнена
- Сложность алгоритма $O(m\,(log\ n + k\,\#seeds_{k=1}))$
  n и m - количество фрагментов в референсе и карте
  k - длина k-tuple
  $\#seeds_{k=1}$ - количество сидов найденных по первому элементу

OPTIMA: Парное выравнивание карты с референсом

После обнаружения схожих сидов на референсе происходит парное выравнивание алгоритмом динамического программирования:

$$Score_{s,t} = \min_{k \le s, l \le t} C_{ce}\,(s - k + t - l) + \chi^2_{k\ldots s, l\ldots t} + Score_{k-1,l-1}$$

$$\chi^2_{k\ldots s, l\ldots t} = \frac{\left(\sum_{i=k}^{s} o_i - \sum_{j=l}^{t} r_j\right)^2}{\sum_{j=l}^{t} \sigma_j^2}$$

C_{se} - штраф за пропущенные разрезы

OPTIMA: Определение значимости выравнивания

Пусть a - выравнивание из множества выравниваний $\mathcal{A}$

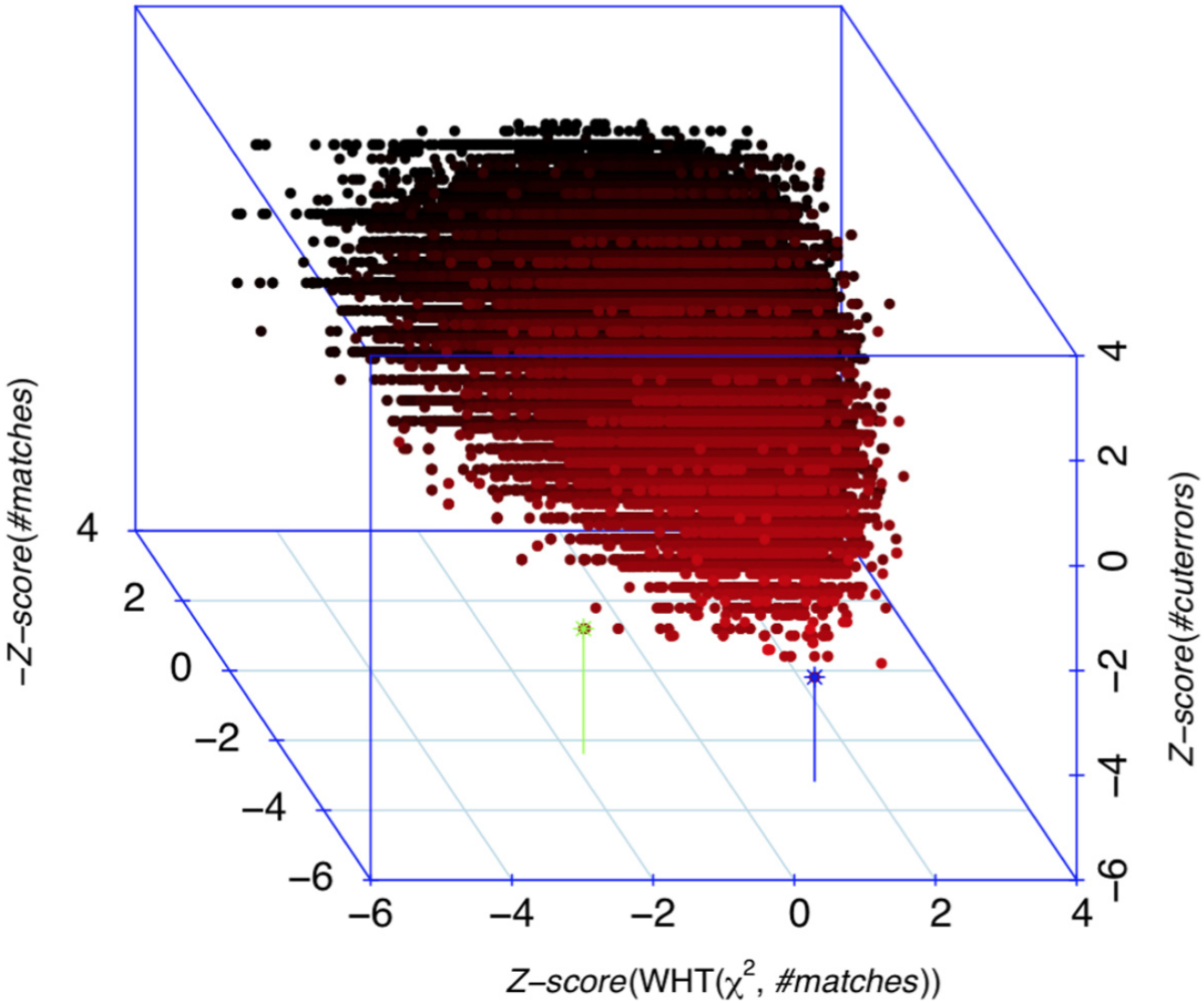
$$Z_{score}(a \in \mathcal{A}, f) = \frac{f_a - Mean(f_{\mathcal{A}})}{SD(f_{\mathcal{A}})}$$

где f - характеристика выравнивания.
Тогда статистическая значимость выравнивания:

$$\begin{aligned}\vartheta(a \in \mathcal{A}) = Z_{score}(& - Z_{score}(a, \#matches) \\ & + Z_{score}(a, \#cuterrors) \\ & + Z_{score}(a, WHT(\chi^2, \#matches)))\end{aligned}$$

$$\text{где } WHT(\chi^2, \#matches) = \frac{\sqrt[3]{\frac{\chi^2}{\#matches}} - \left(1 - \frac{1}{9}\frac{2}{\#matches}\right)}{\sqrt{\frac{1}{9}\frac{2}{\#matches}}}$$

OPTIMA: Пример множества выравниваний



OPTIMA: Результаты

Результаты для 2100 карт:

Algorithm	*Drosophila* (A)		*Drosophila* (B)		Human (A)		Human (B)	
	S	P	S	P	S	P	S	P
OPTIMA	**90**	**100**	**49**	**99**	**83**	**100**	**43**	**98**
Gentig v.2 (d)	59	**100**	24	**99**	53	96	20	80
Gentig v.2 (tp)	59	**100**	24	98	54	95	20	88
SOMA v.2 (v)	72	73	31	39	50	50	17	20
Likelihood (d+a)	49	49	29	30	24	24	14	14
Likelihood (d+a+t)	64	65	38	39	33	34	18	19
Likelihood (p+a+t)	75	75	39	39	62	62	19	20

S - чувствительность

P - точность

tp - настройка параметров в соответствии с генерацией данных

p - параметры, указанные в статьях авторов

d - стандартные настройки

t - обрезание концов карт

a - скорректированные на основе анализа организма

OPTIMA: Время работы

Ожидаемое время работы:

Algorithm	Complexity		Running time	
	Time	Space	*Drosophila*	Human
OPTIMA	$O((m-c)\,\delta^3\,\#seeds)$	$O((m-c)^2 + c\,n)$	**54 m**	**36 days**
Gentig v.2 (d)	$O(\#it\,m\,\delta^3\,\#hashes)$	$O(m^2 + n + \lvert HashTable\rvert)$	1.32 h	75 days
Gentig v.2 (tp)			1.85 h	174 days
SOMA v.2 (v)	$O(m^2\,n^2)$	$O(m\,n)$	1.28 years	1,067 years
Likelihood (d+a)	$O(m\,n\,\delta^2)$	$O(m\,n)$	22.22 h	2.72 years
Likelihood (d+a+t)			19.62 h	2.38 years
Likelihood (p+a+t)			41.73 h	5.53 years

Drosophila - 82000 карт

Human - 2100000 карт

OPTIMA: Вывод

Плюсы:

- Использование композитных сидов
- Определение значимости выравнивания при отсутствии априорной информации о модели ошибок

Минусы:

- Этап хешинга возвращает довольно много стартовых мест
- Долгое время работы

MAligner

MAligner: Общие сведения

Два подхода:

1. На основе алгоритма Смита-Ватермана
   - Построение множества выравниваний на референсе
   - Определение значимых выравниваний по M-Score
2. На основе индексации

MAligner: Алгоритм динамического программирования

Пусть имеются два выравненных участка длины r и q с пропущенными n и m разрезами на референсе и карте соотвественно. Тогда выравнивание имеет следующее значение:

$$Score(q, r, m, n) = S(q, r) + C_q\, m + C_r\, n$$

$$S(q, r) = \left(\frac{q - r}{\sigma(r)}\right)^2$$

$$\sigma(r) = \max(\alpha\, r, \sigma_{min})$$

C_q - штраф за пропущенные разрезы на карте
C_r - штраф за пропущенные разрезы на референсе
σ_{min} - для фрагментов малой длины ошибка больше
α - доля референса, которая будет использовать как стандартное отклонение

MAligner: M-Score - значимость выравнивания

Предложена оценка M-Score для определения значимости выравнивания:

$$m_{\mathcal{A}} = \underset{A \in \mathcal{A}}{median}\{Score(A)\}$$

$$MAD_{\mathcal{A}} = \underset{A \in \mathcal{A}}{median}\{|Score(A) - m_{\mathcal{A}}|\}$$

$$\text{M-Score}_{\mathcal{A}}(A) = \frac{Score(A) - m_{\mathcal{A}}}{MAD_{\mathcal{A}}}$$

$Score(A)$ - значение выравнивания A
$\mathcal{A}$ - 100 лучших выравниваний по Score(A)

MAligner: Алгоритм на основе индексов

Работает в предположении, что в карте не могут быть пропущенные разрезы:

1. Выбирается k и строятся всевозможные k-tuple на референсе длины меньше k - $\mathcal{K}$ и сортируем его по длине.
2. Далее строится по множеству $\mathcal{K}$ граф, где вершины - k-tuple, а рёбрами соединяем те k-tuple, у которых граничные разрезы совпадают.
3. У входной карты берём k-tuple и бинарным поиском по длине в $\mathcal{K}$ ищем схожие k-tuple.
4. Для каждого найденного k-tuple запускаем поиск в ширину по графу, при чём идём только по тем вершинам, длины которых C удовлетворяют очередному фрагменту карты длины c_q:

$$c_q - max(\alpha\, c_q, \beta) \leq C \leq c_q + max(\alpha\, c_q, \beta)$$

5. Получаем набор выравниваний.

MAligner: Результаты

Данные без ошибок:

Software	Total Alignments	Contigs with Alignment	Contigs with Correct Alignment	Contigs with Unique Alignment	Contigs with Unique & Correct Alignment	Runtime
TWIN	37 (2.99 Mb)	31 (2.81 Mb)	31 (2.81 Mb)	28 (2.71 Mb)	28 (2.71 Mb)	0.47s
SOMA	28 (2.66 Mb)	28 (2.66 Mb)	11 (1.48 Mb)	28 (2.66 Mb)	11 (1.48 Mb)	14.22s
malignerIX	36 (2.93 Mb)	31 (2.81 Mb)	31 (2.81 Mb)	29 (2.75Mb)	29 (2.75 Mb)	0.03s
malignerDP	39 (3.01 Mb)	31 (2.81 Mb)	31 (2.81 Mb)	27 (2.68 Mb)	27 (2.68 Mb)	0.40s

Данные с ошибками:

Software	Total Alignments	Contigs with Alignment	Contigs with Correct Alignment	Contigs with Unique Alignment	Contigs with Unique & Correct Alignment	Runtime
TWIN	101 (3.44 Mb)	11 (0.49 Mb)	7 (0.26 Mb)	4 (0.23 Mb)	0 (0.00 Mb)	0.76s
SOMA	6 (0.25 Mb)	6 (0.25 Mb)	0 (0.00 Mb)	6 (0.25 Mb)	0 (0.00 Mb)	14.45s
malignerIX	81 (3.23 Mb)	15 (0.99 Mb)	13 (0.87 Mb)	7 (0.65 Mb)	5 (0.53 Mb)	0.15s
malignerDP	208 (8.35 Mb)	28 (2.37 Mb)	26 (2.24 Mb)	13 (1.60 Mb)	13 (1.60 Mb)	0.31s

MAligner: Вывод

Плюсы:

- Определение значимости выравнивания при отсутствии априорной информации о модели ошибок
- Предложенная оценка значимости является робастной оценкой
- Алгоритм на основе индексов применим, так как количество линших разрезов довольно мало

Минусы:

- Отсутствие этапа хешинга

OMBlast

OMBlast: Алгоритм

Этапы выравнивания:

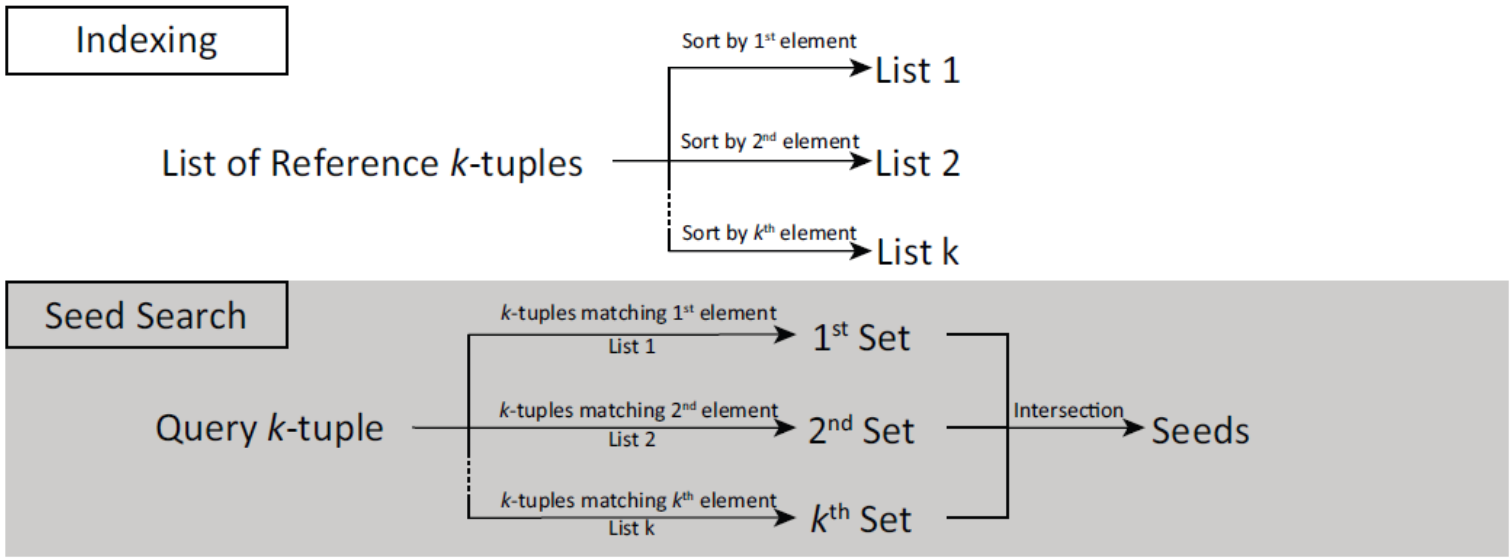
- Поиск стартовых мест (сидов) для начала выравнивания
- Расширение сидов
- Объединение пересекающих выравниваний
- Построение итогового выравнивания

OMBlast: Поиск стартовых сидов - индексация

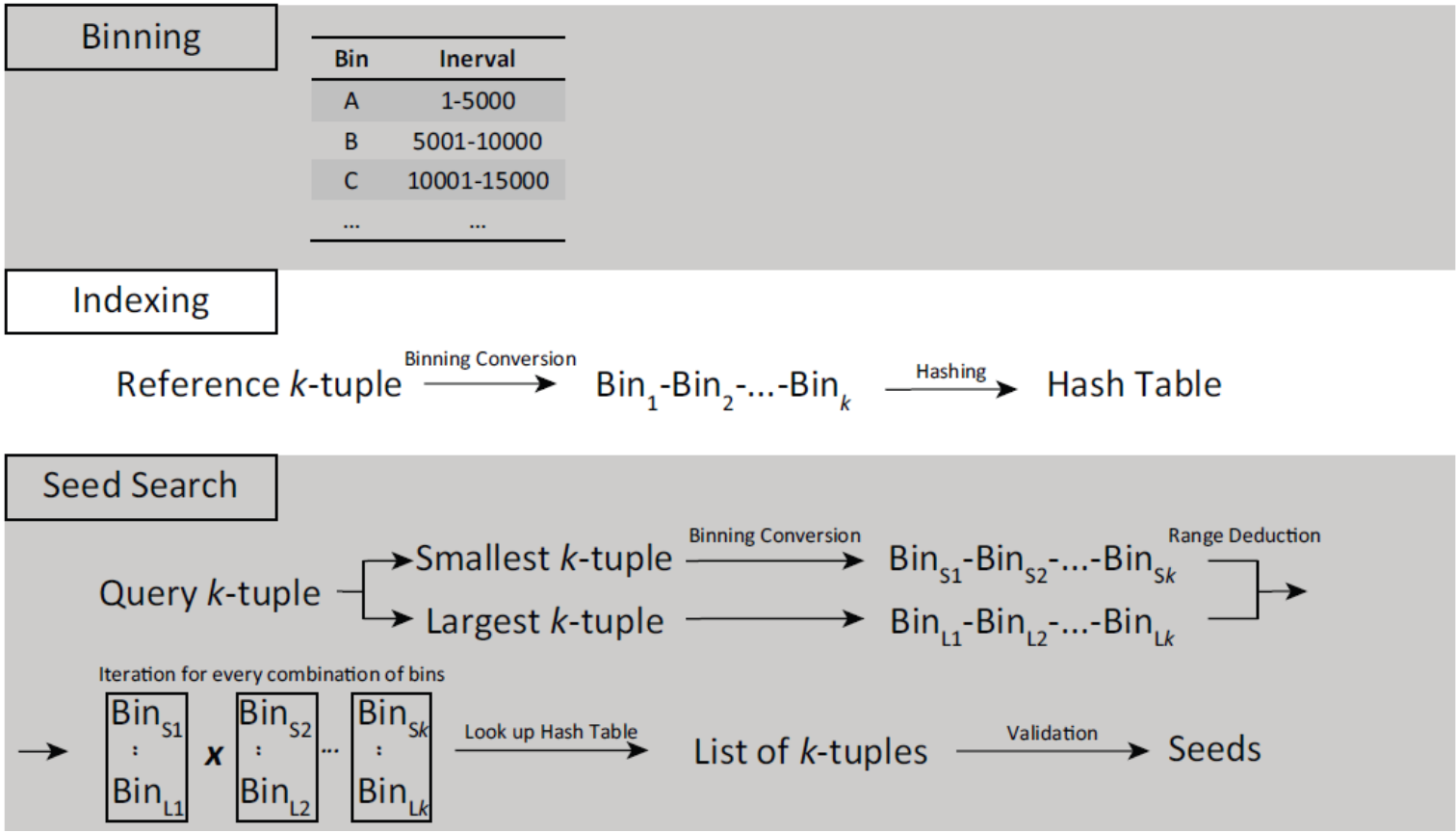
Фрагмент q на карте совпадает с фрагментом r на референсе:

$$r(1 - T_s) - T_m \leq q \leq r(1 + T_s) + T_m$$

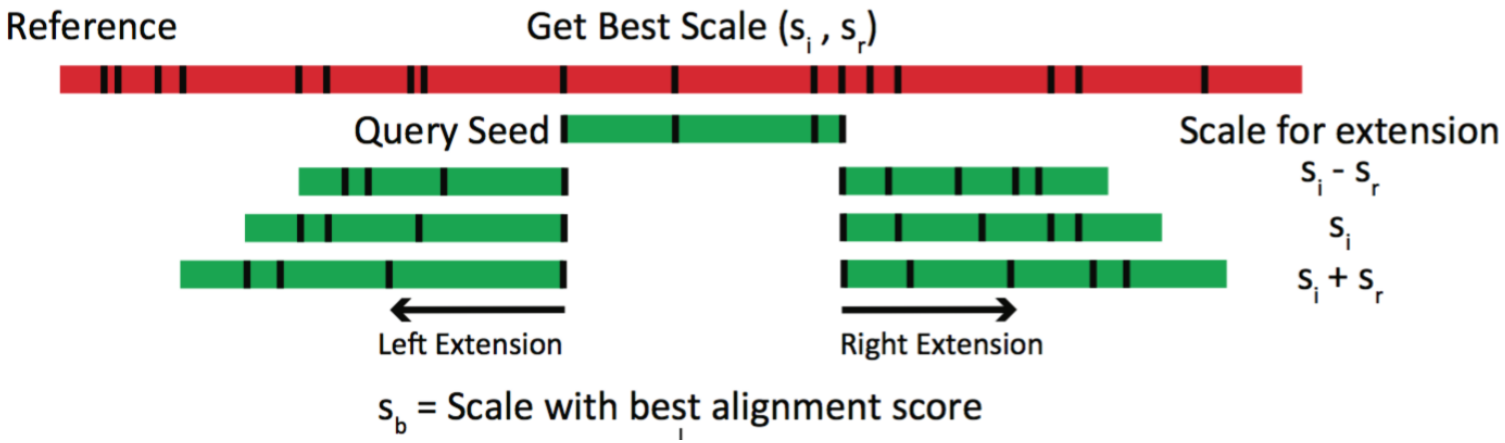
T_s - параметр, ошибка масштабирования
T_m - параметр, ошибка измерений

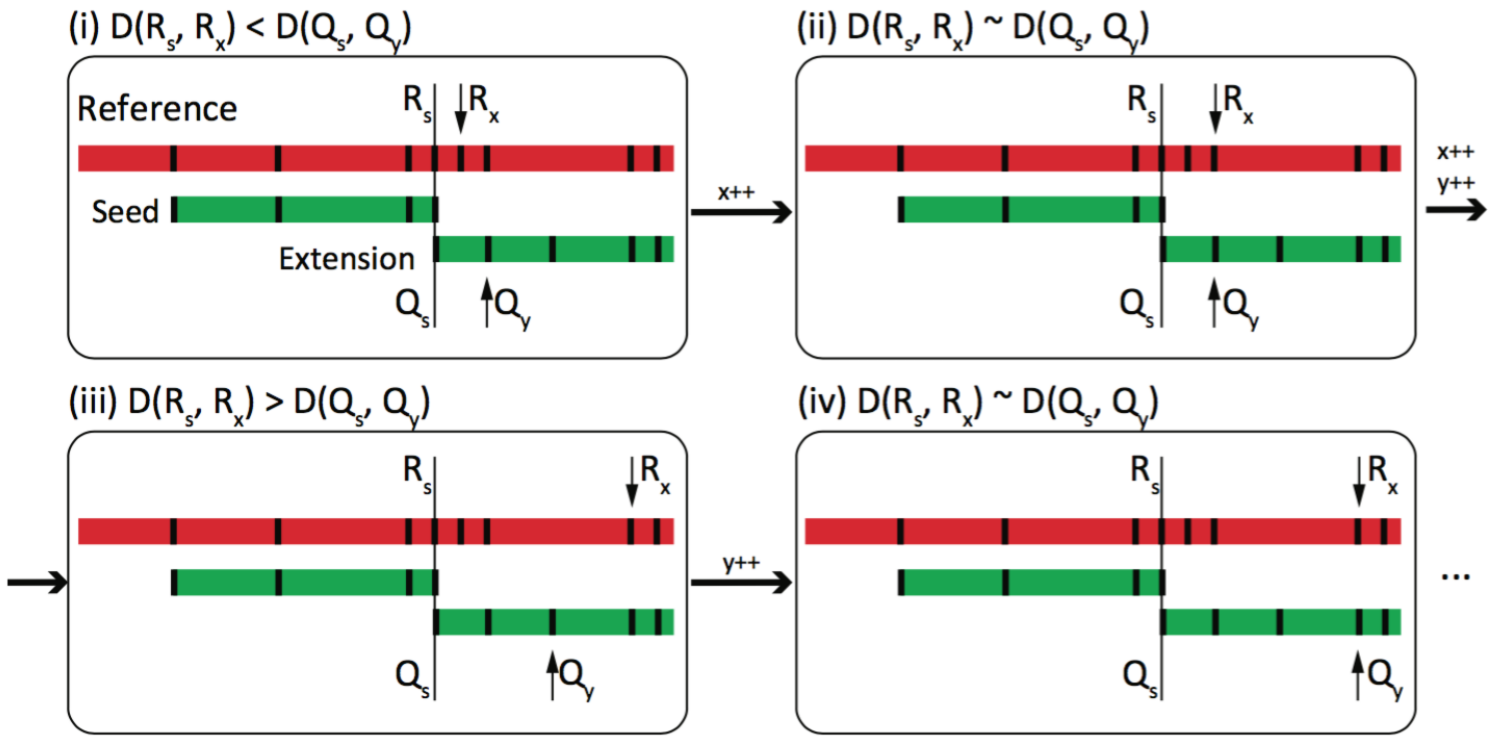


OMBlast: Поиск стартовых сидов - бины



OMBlast: Расширение сидов

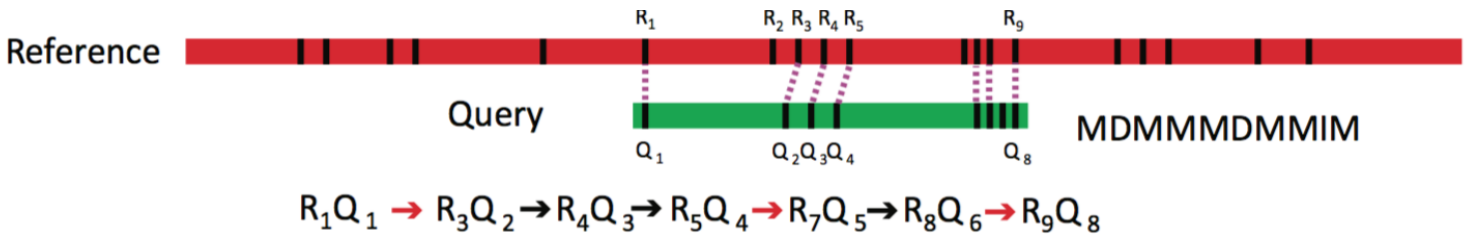




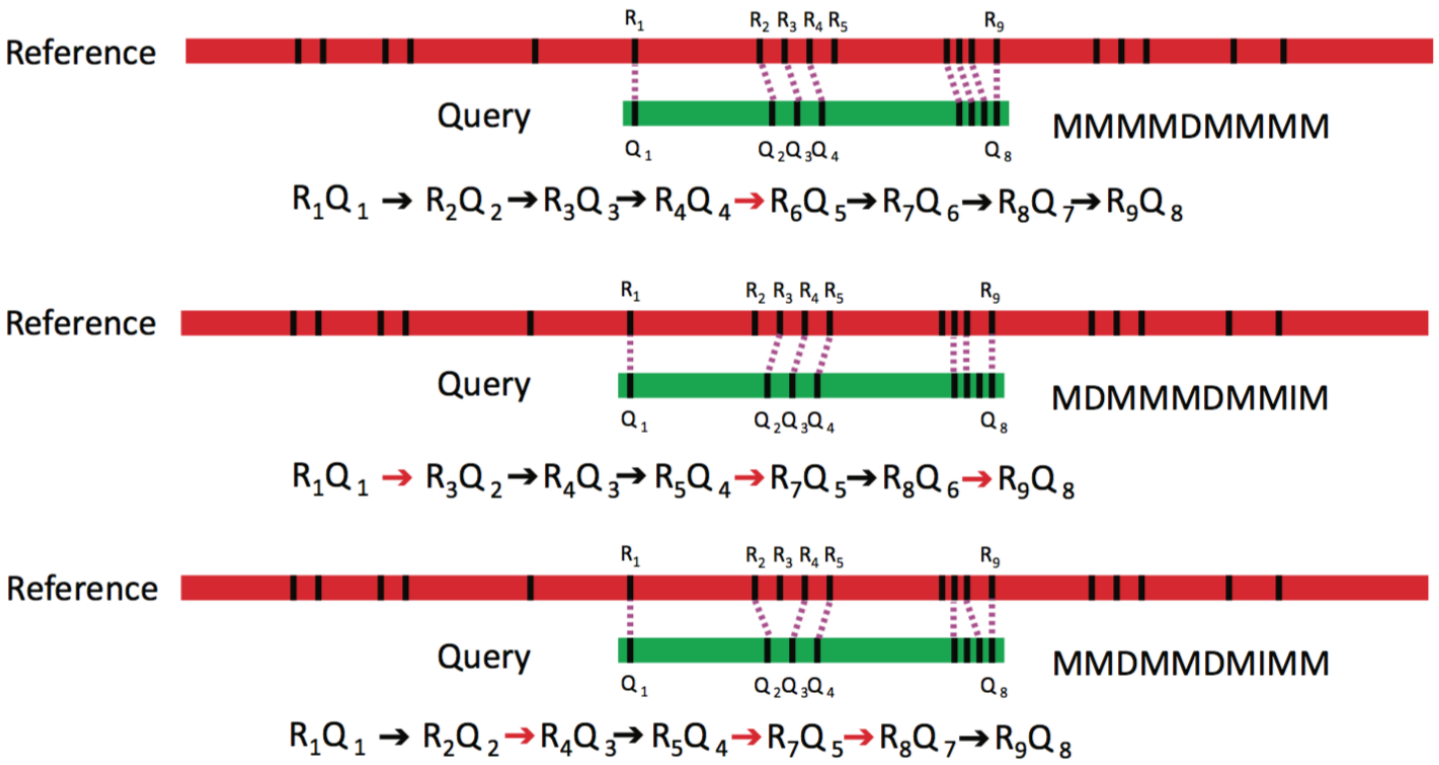
OMBlast: Объединение выравниваний (1)

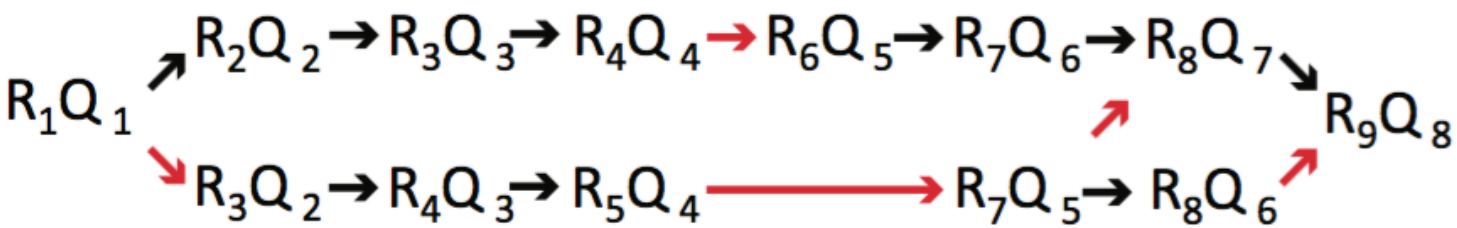
Строится взвешенный ациклический граф:

- Вершины - выравненные разрезы
- Рёбра - между двумя парами последовательно (на одной карте) выравненных разрезов
- Веса - $t_m\, u_m - t_{es}\, u_{es} - t_{ms}\, u_{ms}$
  u_m - количество совпадений
  u_{es} - количество лишних разрезов
  u_{ms} - количество пропущенных разрезов



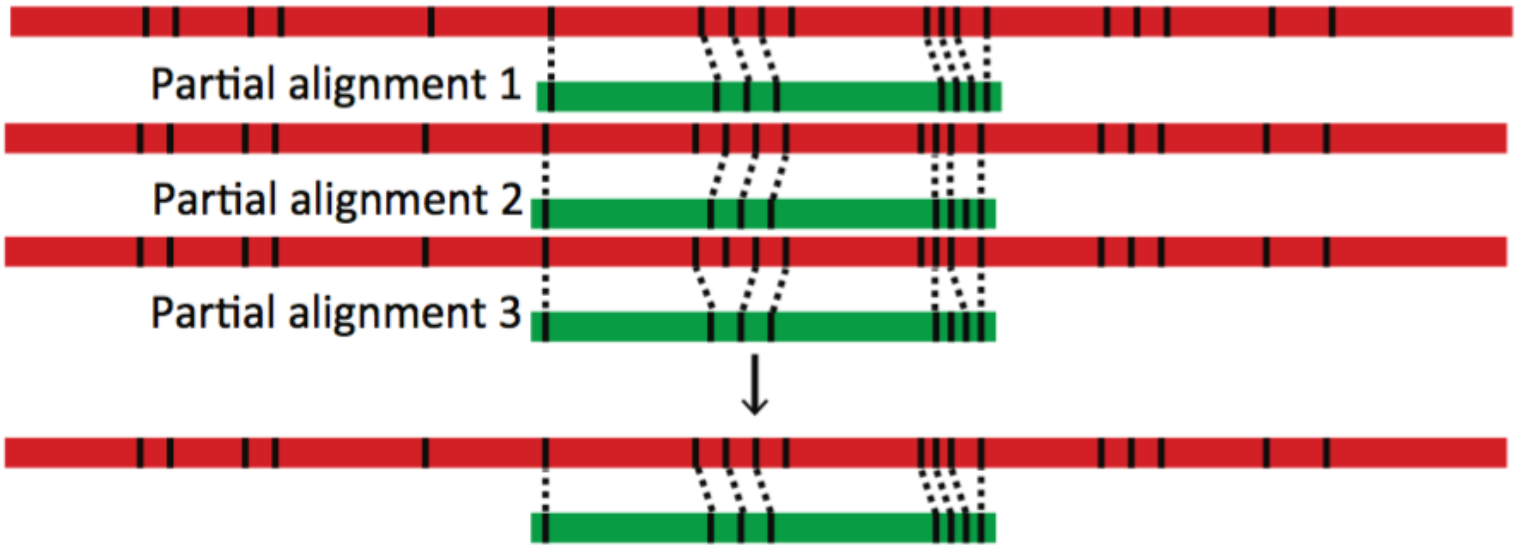
OMBlast: Объединение выравниваний (2)





OMBlast: Объединение выравниваний (3)

С помощью динамического программирования определяется путь в графе с наибольшим весом



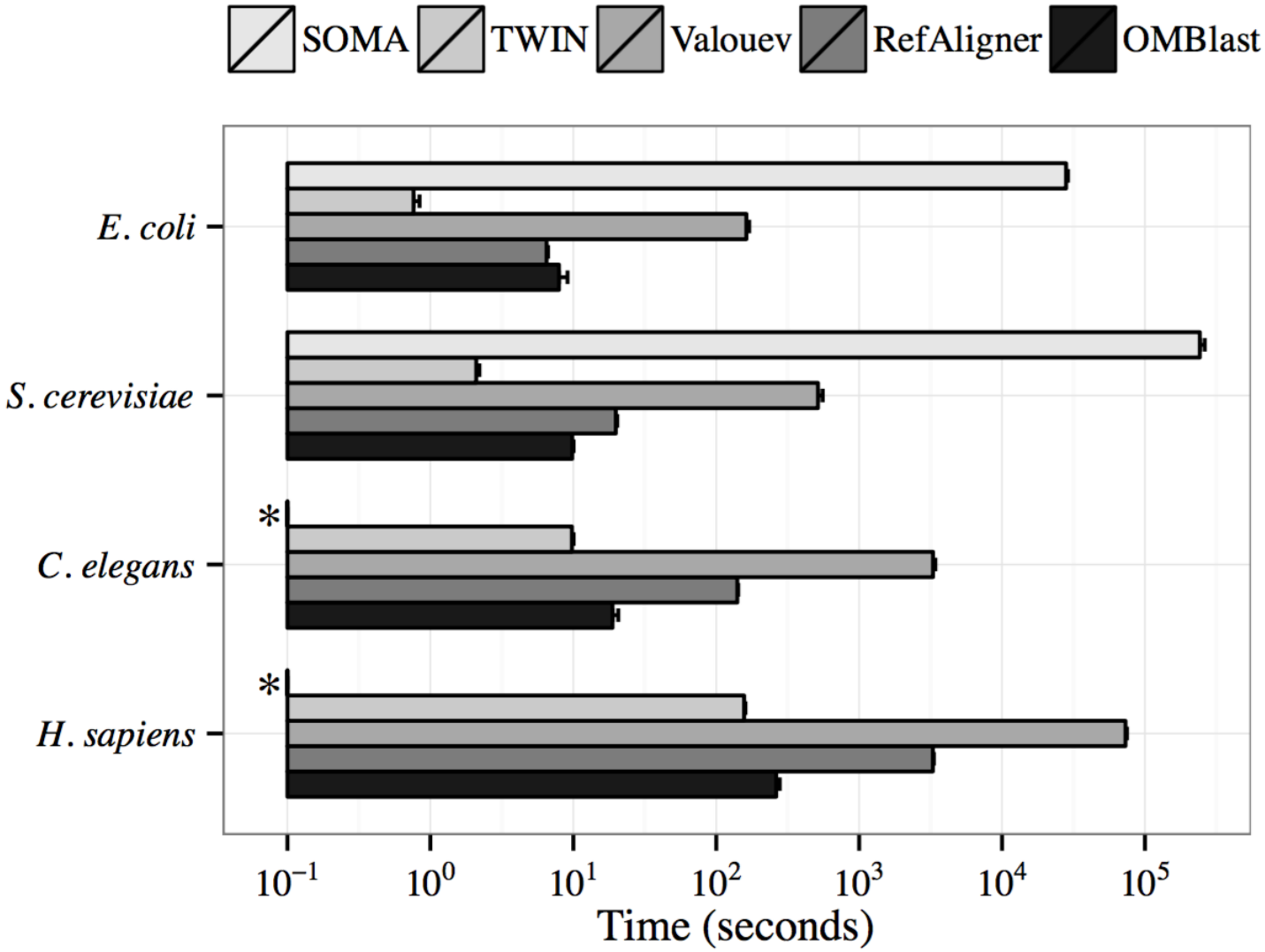
OMBlast: Построение итогового выравнивания

OMBlast: Результаты - входные данные

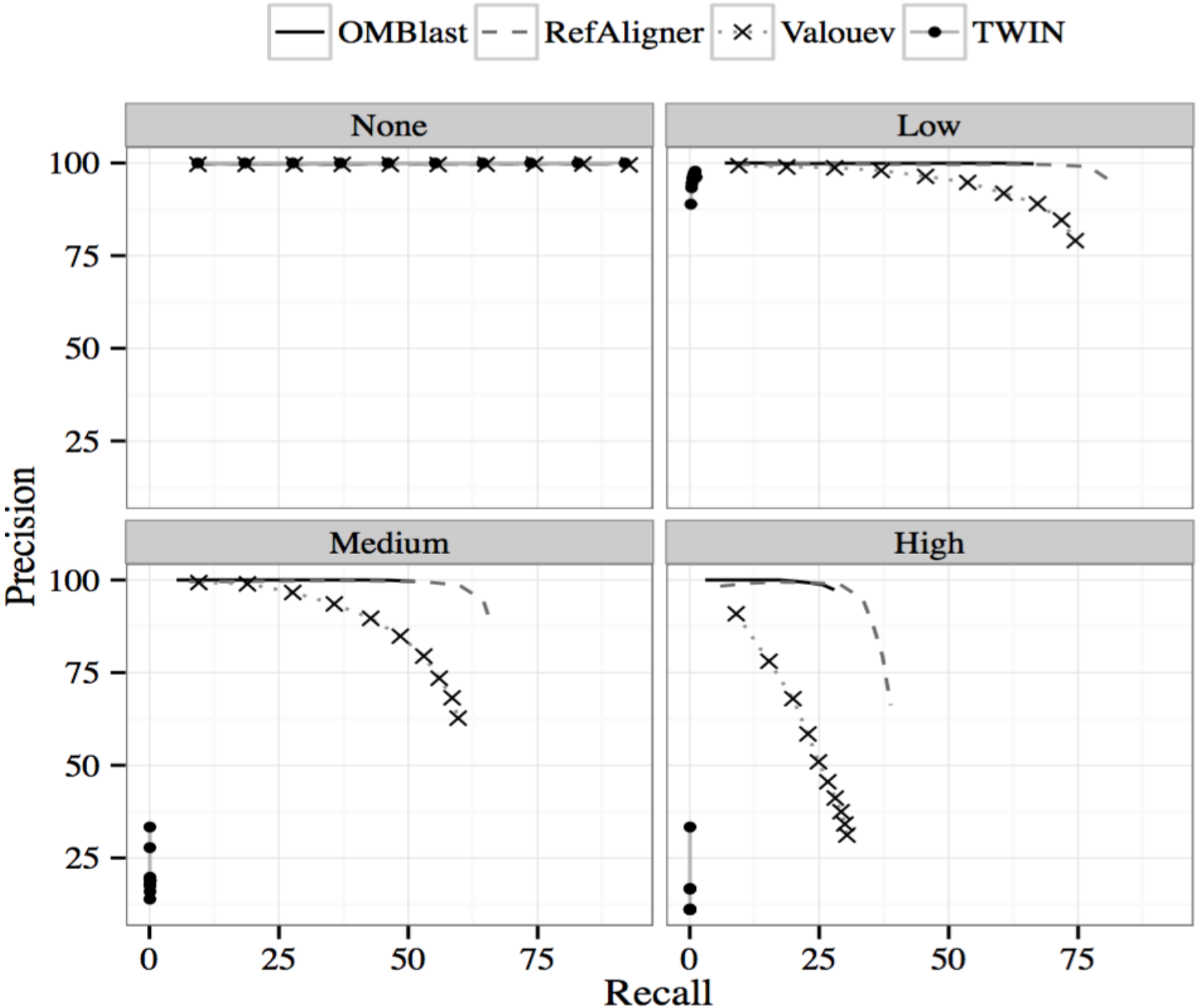
Organism	Genome Size (Mbp)	Total Signals	Average Bases Between Signals (kbp)
E. coli	4.6	683	6.8 ± 7.3
S. cerevisiae	12.1	1953	6.2 ± 6.7
C. elegans	100.3	14837	6.8 ± 8.0
H. sapiens	3088.3	377143	8.2 ± 83.2

Error Rate	None	Low	Medium	High
Extra Signal Rate	0	0.000005	0.00001	0.00002
Missing Signal Rate	0	0.05	0.1	0.2
Scaling	0	0.02	0.04	0.08
Measurement (bp)	0	500	500	500
Resolution (bp)	0	1200	1200	1200

OMBlast: Результаты - время работы



OMBlast: Результаты - точность и полнота



OMBlast: Вывод

Плюсы:

-

Минусы:

- Предложенные схемы хешинга будут занимать много места

TWIN

TWIN: Алгоритм

Алгоритм разработан в предположении отсутствия пропущенных и лишних разрезов. Идея:

- Построение FM-индекса на референсе
- Выравнивание карты на референсе будем искать как подстроку в строке
- В статье предлагается алгоритм неточного поиска подстроки в строке

Ссылки

Ссылки: исходники

В открытом доступе:

- TWIN
- OPTIMA
- MAligner
- OMBlast

Ссылки: статьи I

Alden King-Yung Leung и др. “OMBlast: Alignment Tool for Optical Mapping Using a Seed-and-extend Approach”. в: *Bioinformatics* (). DOI: 10.1093/bioinformatics/btw620.

Menglu Li и др. “Towards a More Accurate Error Model for BioNano Optical Maps”. в: Springer International Publishing, 2016. DOI: 10.1007/978-3-319-38782-6_6.

Lee M. Mendelowitz, David C. Schwartz и Mihai Pop. “Maligner: a fast ordered restriction map aligner”. в: *Bioinformatics* (). DOI: 10.1093/bioinformatics/btv711.

Martin D. Muggli, Simon J. Puglisi и Christina Boucher. “Efficient Indexed Alignment of Contigs to Optical Maps”. в: Springer Berlin Heidelberg, 2014. DOI: 10.1007/978-3-662-44753-6_6.

Ссылки: статьи II

Niranjan Nagarajan, Timothy D. Read и Mihai Pop. “Scaffolding and validation of bacterial genome assemblies using optical restriction maps”. в: *Bioinformatics* (). DOI: 10.1093/bioinformatics/btn102.

Davide Verzotto и др. “OPTIMA: sensitive and accurate whole-genome alignment of error-prone genomic maps by combinatorial indexing and technology-agnostic statistical analysis”. в: *GigaScience* (). DOI: 10.1186/s13742-016-0110-0.

Спасибо за внимание!