

Перспективы низкоуровневой оптимизации современных выравнивателей

Выполнил: А. Диевский

Научный руководитель:
О. Медведев

Напоминание

- Необходимость в быстром выравнивании
- BWT-based / seed-based
- Доступ к RAM – узкое место
- Естественная идея – использование кэша или аппаратное ускорение

BWT-based

- bowtie, BWA etc.
- Не слишком эффективны для неточных прикладываний
- Непредсказуемый доступ к RAM
- Большой индекс

BWT-based

- Оптимизировать доступ к памяти с помощью менеджера доступа
- Это подходит для распараллеливания
- Аппаратная реализация натывается на проблемы доступа к большому индексу
- Возможное аппаратное решение – разбивка генома на маленькие куски; эффективность сомнительна

Seed-based

- BLAST, MAQ, PASS, GASSST etc.
- Три стадии: *seed-filter-extend*
- При разумной реализации большая часть обработки в *filter* происходит с небольшим объёмом данных
- Доступ к индексу гораздо более предсказуем, эвристические таблицы помещаются в кэш

Seed-based

- Доступ к памяти достаточно оптимален
- Возможна аппаратная реализация *filter*-эвристик
- Возможна, наоборот, аппаратная реализация *extend*-выравнивания

Seed-based

- *filter*: в случае неточного выравнивания фильтр перестаёт быть узким местом
- *extend*: отлично распараллеливается, но требует чтения reference-последовательности; здесь узкое место – время чтения, а не время выравнивания
- Впрочем, некоторые бактериальные геномы помещаются в onboard RAM.

Summary

BWT-based:

- менеджер доступа к памяти для распараллеливания

Seed-based:

- аппаратная реализация эвристик для точных выравниваний;
- аппаратная реализация NW для небольших геномов