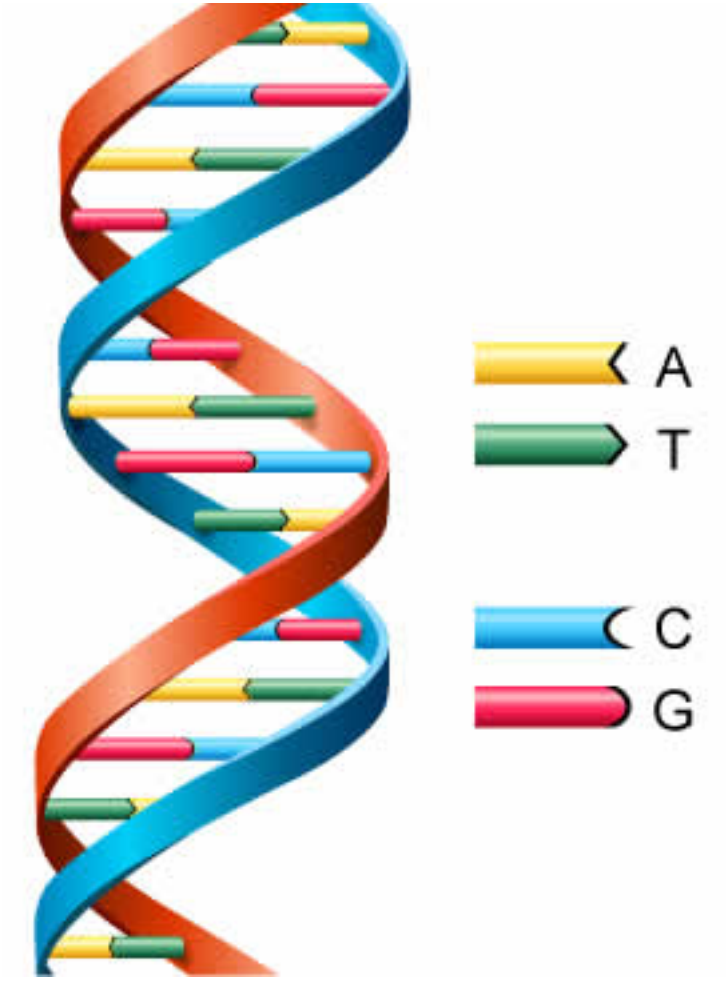
DNA compression

Афанасьев Антон

Руководитель: Вяххи Николай

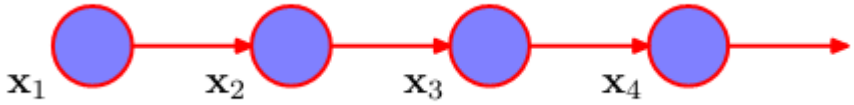
Задача

Исследовать возможности сжатия ДНК-ридов и реализовать прототип программы сжатия



Детали

- Три вида информации: read (sequence), quality, header
- Основная трудность – quality
  - Анализируются вероятности $p(x_n = a_n | x_{n-1} = a_{n-1})$



  - По ним выполняется кодирование (код Хаффмана)
- Для sequence'ов используется кодирование 2мя битами
- Для header'ов используется шаблон, хранится только разница

Задача: по данному множеству "похожих" строк найти "минимальный" шаблон, который match'ит все строки множества.

Пример:
@ERR001268.1 080821_HWI-EAS301_0002_30ALBAAXX:1:1:1115:2003/1
@ERR001268.2 080821_HWI-EAS301_0002_30ALBAAXX:1:1:1090:1998/1
...
@ERR001268.13518977 080821_HWI-EAS301_0002_30ALBAAXX:1:100:417:1009/1

Шаблон:
@ERR001268.(\d*) 080821_HWI-EAS301_0002_30ALBAAXX:1:(\d*):(\d*):(\d*)/1

Кроме того, для численных участков выводится дополнительная информация: число бит, достаточных для хранения, целесообразность хранения разности вместо значений.

Сравнение

Name	ERR001268	ERR003978_1	human
Size	164890754	3471876784	196479831
gzip	57666589	1243926771	70008644
bzip2 -9	47770272	1041237490	56480283
dna_compress	45173761	979537345	55802217

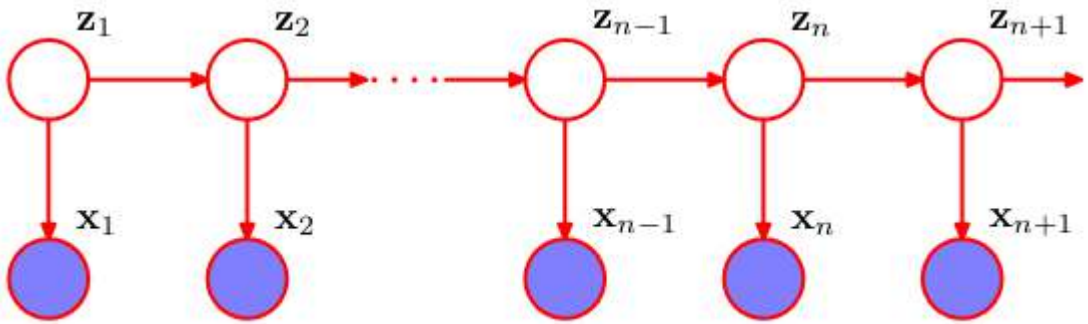
- Написан прототип программы и инструмент для автоматического тестирования

File name	Size before	Size after	Ratio	CTime	DTime	Result
../data/tmp.fastq	139350 bytes	104082 bytes	74%	1.82s	0.39s	Passed
../data/tmp2.fastq	1405660 bytes	470515 bytes	33%	7.18s	2.35s	Passed
../data/tmp3.fastq	14139653 bytes	3259584 bytes	23%	17.21s	20.37s	Passed

- Исследовано несколько вариантов сжатия
- Проведенено сравнение с готовыми реализациями
- Код выложен тут: http://github.com/bioinf/dna_compression

Возможности

- Рефакторинг и ускорение (Python -> C++)
- Поддержка различных форматов fastq
- Референсное сжатие
- Использование арифметического кодирования вместо Хаффмана
- Lossy варианты
- Hidden Markov Model для предсказания?



Спасибо за внимание!