Reference Based Genome Compression

B. G. Chern, I. Ochoa, A. Manolakos, A. No, K. Venkat and T. Weissman
Information Systems Laboratory, Stanford University

Introduction:

DNA sequencing technology has advanced to a point where storage is becoming the central bottleneck in the acquisition and mining of more data. Large amounts of data are vital for genomics research, and generic compression tools, while viable, cannot offer the same savings as approaches tuned to inherent biological properties.

We propose an algorithm to compress a target genome given a known reference genome. The genomes are related via insertions, deletions and substitutions. The proposed algorithm first generates a mapping from the reference to the target genome, and then compresses this mapping with an entropy coder.

Problem description:

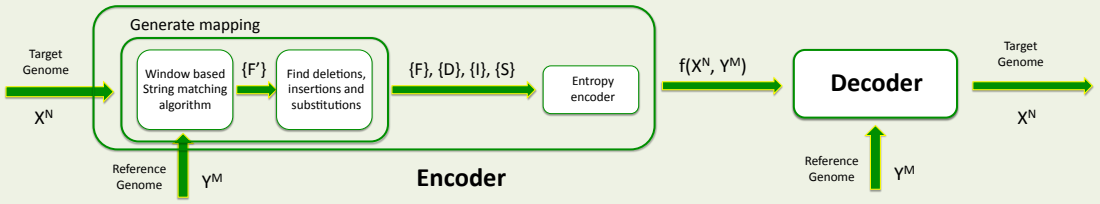
We consider the **problem** of compressing the target genome X^N given the reference genome Y^M as side information, which is available at both the encoder and decoder.

The **encoder** is described by the mapping $f(X^N, Y^M)$, which indicates the compressed version of X^N given Y^M.

The **decoder** is described by the mapping $g(f(X^N, Y^M), Y^M)$, which satisfies $X^N = g(f(X^N, Y^M), Y^M)$.

The **objective** is to construct an encoding/decoding scheme that minimizes the length of the representation f(., .), while ensuring a perfect reconstruction of X^N at the decoder.

Proposed scheme:



1) Window based string matching algorithm (based on LZ77)

Idea: starting from a position in the target sequence, find the longest matching string within a fixed window in the reference sequence.

More formally: Define $y_{W_k - L_k}^{W_k + R_k}$ as the window at time k. Assume $x_1^{n_k}$ has already been encoded, for some n_k. Then, find the largest length l_k s.t. $x_{n_k+1}^{n_k+l_k} = y_i^{i+l_k-1}$ and $x_{n_k+1}^{n_k+l_k+1} \neq y_{i+l_k}$, for some $W_k - L_k \le i \le W_k + R_k$ and store the position $p_k = i$, length l_k and the novel symbol $z_k = x^{n_k+l_k+1}$. Set $n_{k+1} = n_k + l_k + 1$ and repeat the procedure until the target genome is exhausted.

Notice that the set of instructions F = {(p, l, z)} suffices to reconstruct the target genome given the reference.

The window is dynamically updating its position and width along the reference so as to efficiently capture the synchronization in traversing along the two sequences.

Example:

REF: $A_1ACCT_5GGCAT_{10}TGCAA_{15}CTTAA_{20}CTACC_{25}CGTTC_{30}AATCC_{35}CACTA_{40}\cdots$
TARGET: $A_1ACCG_5GGCAT_{10}TGTTT_{15}AACGT_{20}ACCCG_{25}TTAAT_{30}CCCCA_{35}CTT_{40}\cdots$

F = { (1, 4, G), (6, 7, T), (17, 5, G), (22, 8, A), (32, 5, C), (37, 3, T), ...}

2) Find substitutions, insertions and deletions and modify F accordingly

Substitutions: $S = \{(p^{(s)}, z^{(s)})\} \rightarrow S = \{(5, G)\}$
Insertions: $I = \{(p^{(i)}, z^{(i)})\} \rightarrow I = \{(19, G), (34, C)\}$
Deletions: $D = \{(p^{(d)}, l^{(d)}, z^{(d)})\} \rightarrow D = \{(27, 1)\}$
Instructions: $F = \{(p, l, z)\} \rightarrow F = \{(1, 12, T), (17, 23, T)\}$

3) Entropy encoder for sets F, S, I and D

Techniques: Delta encoding, Huffman, Golomb codes...

Compression Performance:

Data: *hg18* release from the UCSC Genome Browser, the korean genomes *KOREF20090131* and *KOREF20090224*, the genome of a Han Chinese referred to as *YH* and the Watson *JW* genome.

Performance: We compare the proposed scheme with *Gzip* and the algorithm *GReEn* proposed in [2]. All the sizes are expressed in MB.

Ref	Target	Size	Gzip	GReEn	Our
hg18	JW	2,991	834.8	18.23	**6.99**
hg18	YH	2,987	832.0	9.85	**8.50**
KO224	YH	2,987	832.0	10.06	**9.54**
KO331	JW	2,991	834.8	23.23	**10.22**

Acknowledgement: The authors would like to thank Itai Sharon, Chris Miller and Golan Yona for helpful discussions.

References:

[1] B. G. Chern, I. Ochoa, A. Manolakos, A. No, K. Venkat and T. Weissman, "Reference Based Genome Compression", http://arxiv.org/pdf/1204.1912v1.pdf

[2] A. J. Pinho, D. Pratas and S. P. Garcia, "GReEn: a tool for efficient compression of genome resequencing data", Nucleic Acids Research, December 2011.