

The SSAHA Trace Server

Zemin Ning, William Spooner, Adam Spargo, Steven Leonard, Mark Rae, Antony Cox
The Wellcome Trust Sanger Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge CB10 1SA
zn1@sanger.ac.uk

Abstract

We present a client/server database system with the potential to make all DNA sequences searchable. The database estimated to be approximately 200 Gbps, contains various types of sequences, including WGS and clone reads, draft assemblies, finished sequences, refSeq etc. The search engine will be SSAHA2, a package combined SSAHA [1] (Sequence Search and Alignment by Hashing Algorithm) with cross_match [2]. Matching seeds of a few kmer words are detected by the SSAHA algorithm. Both query and subject sequences are cut off according to the locations of the matching seeds and then passed to cross_match for full alignment. We aim to develop a platform-independent client/server system which can provide a near real-time (under 10 seconds) search service for a clustered database.

1. Introduction

Various genome projects have brought the creation of many large biological databases. The total data size of DNA sequences, for example, is estimated to be approximately 200 GB, including WGS and clone reads, finished sequences, refSeq etc. Designing services to make all the data searchable in a fast, sensitive and flexible way, poses significant challenges in both development of algorithms and hardware architecture implementation. In this paper, we outline a system with the potential to accomplish this challenging but extremely worthwhile task.

2. Sequence encoding

We consider the problem of searching for exact or partial occurrences of a query sequence Q within a database of subject sequences $D=\{S_1, S_2, \dots, S_d\}$. Each sequence in D is labelled with an integer i which we refer to as its index. We use the term k -tuple to denote a contiguous sequence of DNA bases that is k bases long. A sequence S of DNA that is m bases long will contain $(m-k+1)$ k -tuples. The offset of a k -tuple within S is the position of its first base with respect to the first base of S . We use the letter j to denote offsets and use the notation $w_j(S)$ to denote the k -tuple of S that has offset j . Thus it is

clear that the position within D of each occurrence of each k -tuple may be described by an (i,j) pair. Each of the four possible nucleotides may be encoded as two binary digits as follows:

$$\begin{aligned} f(A) &= 00_2, \\ f(C) &= 01_2, \\ f(G) &= 10_2, \\ f(T) &= 11_2. \end{aligned} \quad (1)$$

Using this encoding, any k -tuple $w=b_1b_2\dots b_k$ may be uniquely represented by a $2k$ bit integer

$$E(w) = \sum_{i=1}^k 4^{i-1} f(b_i) \quad i = 1, 2, \dots, k. \quad (2)$$

3. SSAHA2

SSAHA2 is a package combined SSAHA with phrap/cross_match developed by Phil Green at the University of Washington [2]. Matching seeds, a few exactly matched kmer words are detected from the database by the SSAHA algorithm [1]. SSAHA achieves its fast search speed by converting sequences information into a "hash table" data structure, which can then be searched very rapidly for matches. When the location information of matching seeds is obtained, we then cut off both query and subject sequences and pass the two sequences to cross_match for full alignment. Extra sequences with a given edge length are used for both query and subject to extend the alignment length. In terms of software implementation, cross_match has been imbedded into the SSAHA system and alignment functions are used as libraries.

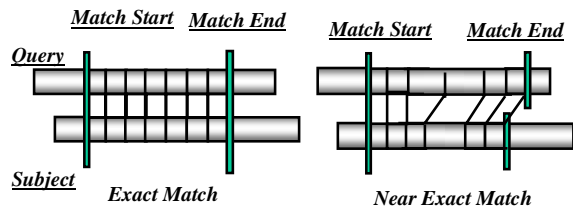


Figure 1 Matching seeds both in query and in subject. In the matching region, gaps are allowed in order to increase sensitivity.

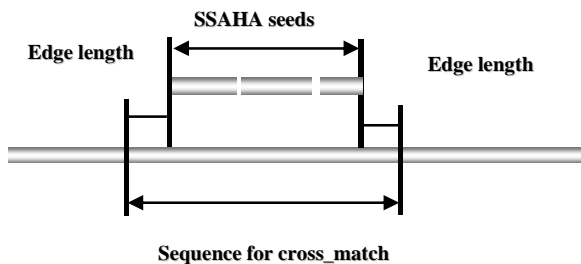


Figure 2 Extra sequences with a given edge length are used to extend the length of alignment.

4. SSAHA2 client/server

SSAHA2 Client:

- (1) Communicates over TCP/IP with the SSAHA2 server;
- (2) Inputs the query data;
- (3) Outputs the alignment results.

SSAHA2 Server:

- (1) Communicates with the SSAHA2 client;
- (2) Receives input query data and carries out search for matching seeds and locations;
- (3) Performs gapped alignment;
- (4) Outputs the search results to the client.

5. Data code and filtration:

In order to speed up sequence search, a system with multiple CPUs is necessary for a database with such big size. We may classify data into three categories:

- Species_Code – Human, mouse, zebrafish, etc;
 Trace_Type – Finished sequence, WGS reads, EST reads, etc;
 Centre_Name – SC, WIBR, WUGSC, etc.

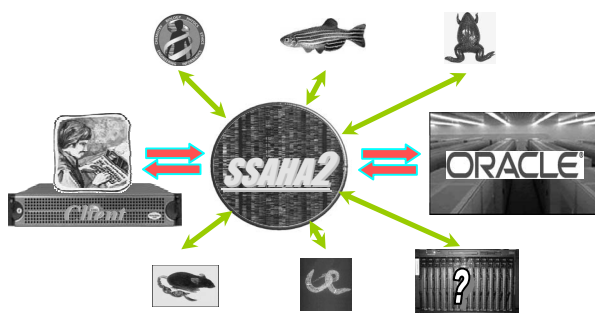


Figure 3 The SSAHA2 client/server system.

Hash tables are generated according to the data categories. Each CPU node is allocated to a given hash

table. Take Mouse genome sequences for example, we may have one hash table for the golden path assembly released by NCBI, one hash table for refSeq, one for EST reads, and two or three hash tables for WGS reads. When a user sends a query for search, he/she might specify the option: mouse assembly Build_32. The query is sent then to the corresponding CPU node which stores the mouse assembly hash table. By doing this, CPU time can be significantly reduced. If no option is given, the query is searched against all the hash tables.

6. Memory and hardware requirement

In the hash table, we store two pieces of information for each k-tuple, sequence index and offset value in the sequence. This means k bases with 8 bytes memory. However, we can combine sequence index and offset into one integer and this reduces to k bases for 4 bytes. We also need memory to store hash table index with a length of 4^k . Therefore, the memory requirement for the system is

$$M = 4 * N_s / k + 4 * 4^k$$

where N_s is the total number of base pairs in the subject database; k is the k-tuple length. When N_s is large enough, memory is about one-third of the database size. For 200 Gbp DNA sequences, we need 70 GB RAM memory. This can be 5 Linux boxes (each with 4 CPUs and 16 GB memory) or 3 boxes with 32 GB RAM memory

A platform-independent client/server code has been developed for data input and alignment output under multiple machines. It is aimed to provide a near real-time (under 10 seconds) search service for a clustered 200 GB database. The solution is also extensible by plugging extra appliances.

7. References

- [1] Ning, Z., Cox, A.J. and Mullikin, J.C. 2001. SSAHA: A Fast Search Method for Large DNA Databases. *Genome Research* 11:1725-1729.
- [2] <http://www.phrap.com/>

Acknowledgement

We would like to thank Professor Phil Green, University of Washington, who has kindly agreed for the use of Phrap/Cross_Match package for sequence alignment in the SSAHA system. The project is funded by the Wellcome Trust.