Performance Study of Genomic Sequence Data Management

The HDF Group

March, 2008

Introduction

Genomic sequence data is typically stored and managed in text files using the FASTA format. Although this allows for convenient processing using scripting languages and text-management tools, as the number of sequences in a file increases, tasks like searching for a particular sequence can have large latencies. Also, the organization of complex data and metadata can be cumbersome when it must be stored and accessed strictly sequentially within a text file. Advanced file formats such as HDF5 can resolve these issues and provide for more efficient data management.

HDF5 is a widely used portable file format and library for storing, retrieving, analyzing, visualizing and exchanging data. HDF5 stores multidimensional arrays along with metadata and it provides a grouping structure that gives users a high degree of flexibility for organizing and managing data. HDF5 also offers storage options, such as compression and chunking that can facilitate efficient data storage and fast access.

As part of our investigations into the use of HDF5 for bioinformatics applications, a prototype Perl package, BioHDF_Perl [1], has been specifically developed to facilitate the storage and management of genomic sequence data in HDF5 format. The BioHDF_Perl User's Guide describes how HDF5 can be used to store the types of DNA sequences and quality values used in this study.

In this brief study, we compare the HDF5 with the FASTA format for storing DNA sequences and quality values. We assume a scenario in which a large collection of sequences are to be stored, and sequences are accessed individually from the collection. DNA sequences and quality values are organized in an HDF5 file for efficient storage and searching, taking advantage of HDF5's compression and chunking options, as well as the ability to supplement the raw data with indexes to facilitate random access. The corresponding FASTA files are unaltered from their standard format.

We compare the two formats in terms of (a) storage use and (b) time to access genomic sequence data using traditional text-management tools for FASTA and BioHDF_Perl for HDF5. Results show that HDF5 can provide storage efficiency through its use of compression and still allow fast random access through its ability to store indexes along with compressed, chunked data.

Testing for Storage Efficiency

Our first test compares the storage space used by genomic sequence data in FASTA and HDF5 formats. In order to obtain relevant numbers for our study, the following set of large FASTA files was downloaded from ftp://ftp.ncbi.nih.gov/pub/TraceDB/13694_environmental_sequence

```
fasta.13694_environmental_sequence.020
qual.13694_environmental_sequence.020
```

The first file contains the sequence bases; the second file contains the corresponding quality values. These files contain a total of 262,778 sequence records and 280,776,775 bases. A Perl script was used to read data from these files and create a sequence collection in an HDF5 file using BioHDF_Perl. The HDF5 file contained all of the information from the original FASTA files, as well as an index to facilitate fast searching. For details and an example of this process, see the BioHDF_Perl User's Guide at [1].

The HDF5 array datasets in the collection were created using the default storage properties, i.e. a data compression level of 7, and storage chunks set to accommodate 5000 array elements each. These values are defined by BioHDF_Perl constants. The resulting file sizes are shown in Figure 1.

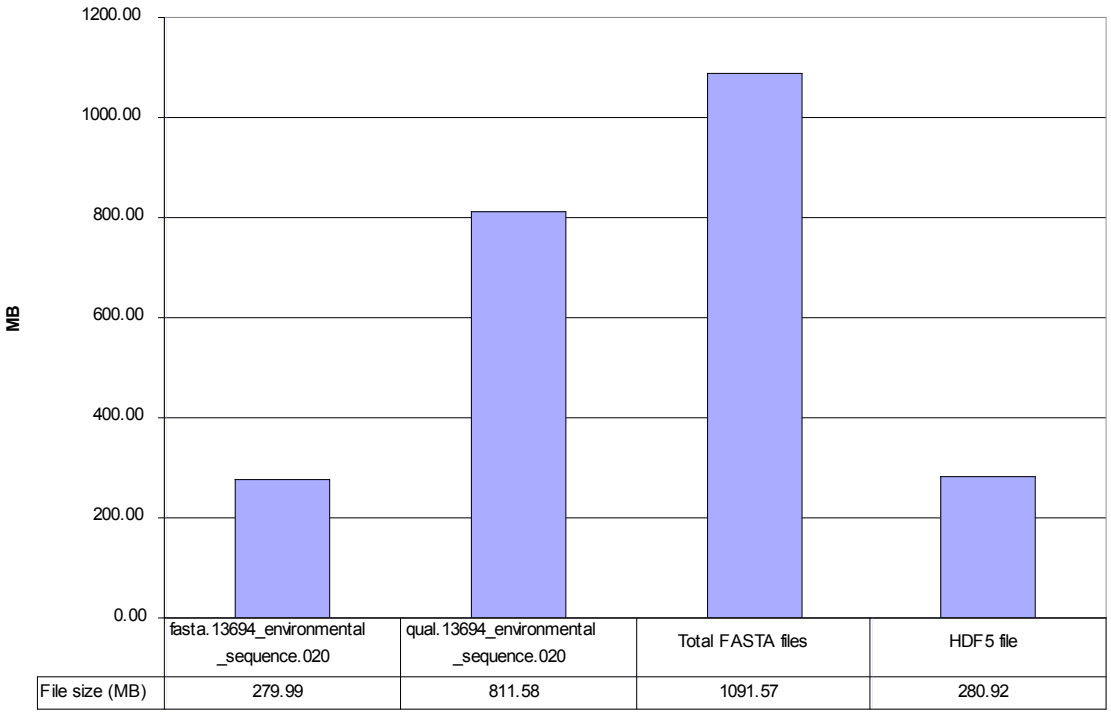


Figure 1 Sizes of files containing genomic sequence data

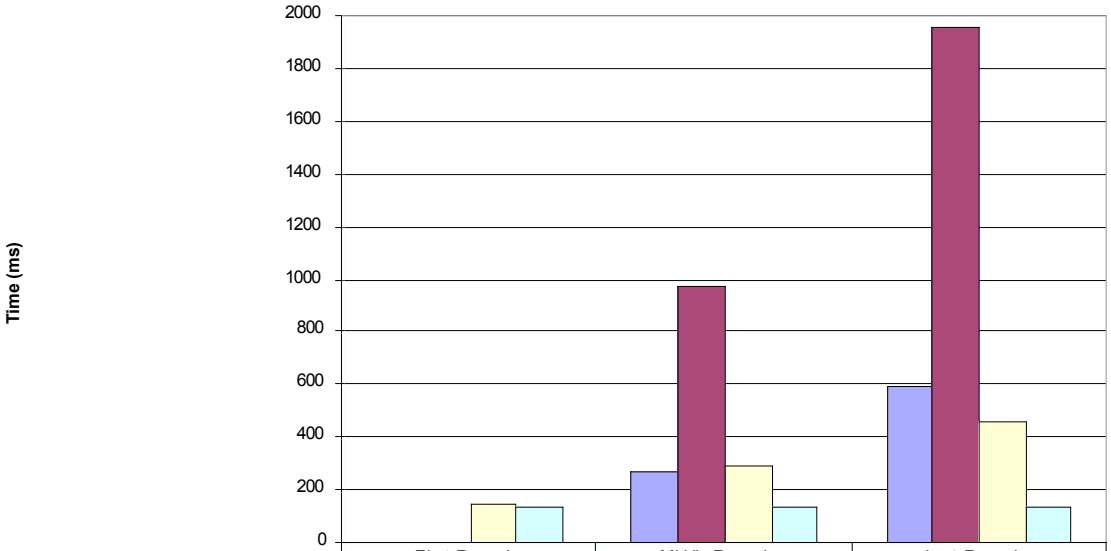
The migration of the data present in both FASTA files into HDF5 format provided a storage space saving of 74% approximately due the application of data compression and better data organization.

Performance Testing

Testing was also carried out to compare access performance when using text-management tools to access a sequence and its quality values in FASTA files and using BioHDF_Perl to access the same data in HDF5. The task was to locate a sequence record using the record identifier. Three cases were considered, in which the requested record was located at the beginning, the middle, and the end of the file. Tests were performed on a Linux Intel 64-bit system with 4GB of memory approximately.

Traditionally, searches are executed using the UNIX command `grep` with the options `-n -m 1` on the FASTA files to obtain the line number and to stop searching after the first instance is found. In our Perl script the same task is accomplished by using the function `BioHDF_Perl::get_sequence` on the HDF5 file. This function employs two search algorithms that are selected automatically depending on the sorting status of the collection with respect to the sequence identifiers. If the collection is not sorted, the function performs a linear search reading 1000 records at a time to reduce access latency (value defined by a BioHDF_Perl constant). On the other hand, if the collection has been sorted using `BioHDF_Perl::sort_sequence_collection`, the search function performs a binary search. The system `time` command was used on the `grep` command and on the Perl script to measure the time to access a record. The results of the performance tests are shown in Figure 2.

The search time results on the first record show that `grep` has minimum overhead and can find the requested sequence faster than BioHDF_Perl. However, as the search space becomes larger, `grep` performance declines while BioHDF_Perl scales very well. In particular, note that the binary search performance remains nominally constant throughout the test cases, indicating that the access time is mostly overhead. Furthermore, while `grep` only locates the line in the file where the record is located, BioHDF_Perl extracts the associated data (comments, bases, and quality values) and places them in Perl structures ready for management and analysis.



	First Record	Middle Record	Last Record
grep on fasta file (ms)	0	280	600
grep on qual file (ms)	0	980	1960
BioHDF_Perl linear search (ms)	150	300	470
BioHDF_Perl binary search (ms)	140	140	140

Figure 2 Times to access a sequence record

Conclusion

Testing was carried out to compare the FASTA format with HDF5 for storing and accessing genomic sequence data. Results indicate that HDF5 saves significant storage space when data compression is applied. Tests also showed that by using HDF5's capability to create indexes and other organizing structures, BioHDF_Perl can perform record searches much faster than is the case using the UNIX tool `grep` when the files contain a large number of sequences.

Reference

[1] "HDF bioinformatics software" web site: http://hdfgroup.org/projects/bioinformatics/bio_software.html.