Hadooping the Genome: The Impact of Big Data Tools on Biology

Hallam Stevens

[Accepted version]

Published in Biosocieties 11: 352-371 (2016)

Link to publishers' version:

https://link.springer.com/article/10.1057/s41292-016-0003-6

Abstract

This essay examines the consequences of so-called "big data" technologies in biomedicine. Analyzing algorithms and data structures used by biologists can provide insight into how biologists perceive and understand their objects of study. As such, I examine some of the most widely used algorithms in genomics: those used for sequence comparison or sequence mapping. These algorithms are derived from powerful tools for text-searching and indexing that have been developed since the 1950s and now play an important role in online search. In biology, sequence comparison algorithms have been used to assemble genomes, process next-generation sequence data, and, most recently, for "precision medicine." I argue that the predominance of a specific set of text-matching and pattern-finding tools has influenced problem choice in genomics. It has allowed genomics to continue to think of genomes as textual objects and to increasingly lock genomics into "big data"-driven text-searching methods. Many "big data" methods are designed for finding patterns in human-written texts. But genomes and other 'omic data are not human-written and are unlikely to be meaningful in the same way.

Introduction

This essay examines the how the use of so-called "big data" technologies in biomedicine has led to the dominance of a specific set of tools in genomics. What happens when tools developed for managing Google's File System end up organizing genomic data? What difference does it make that the same algorithms that run Yahoo and Facebook are also searching biomedical data for evidence of disease? What kinds of questions do these approaches open up or close off to investigation? And what broader socio-economic impacts are these changes in biomedical practice likely to have?

To begin to answer these questions, we need to understand some of the ways in which "big data" itself is beginning to come under scrutiny. "Big data" is a contested term; in this essay I use the term to refer to algorithms and infrastructures used by large Web-based companies (Google, Facebook, Yahoo, and a few others) to manage and analyze their massive volumes of data. The nascent field of "critical data studies" (Mackenzie 2012; Kitchin 2014; Ruppert *et al* 2015; Gitelman 2013; Dalton and Thatcher 2014) has begun to highlight some of the risks and challenges posed by these new technologies. As danah boyd and Kate Crawford have pointed out, "big data" is characterized by arrogance and surrounded with a mythology that it provides access to higher forms of truth: more accuracy and more objectivity (boyd and Crawford 2012, 663). The hype- and rhetoric-powered financial investment in "big data," reinforces the idea that bigger (data) is always better. This has the potential to destabilize ways of knowing that cannot deal with such large data volumes (boyd and Crawford 2012, 666).

I will argue here that in biomedicine, big data tools pose a related but more specific set of risks. Most importantly, they have led to the increasing dominance of a specific set of textual pattern-matching tools in genomics. These tools, while powerful, constrain the kinds of questions and answers that genome biologists pose and attempt to answer. Finding patterns and over-represented correlations within texts has become the one of the primary ways of understanding genomes and what we can do with them. However, many of the algorithms deployed for this work are derived from tools designed to search other kinds of large bodies of text (especially the World Wide Web). This has meant that some of the most important ways in which biologists attempt to understand genomes are now deeply intertwined with ways of searching and understanding the Web. Ultimately, this may limit the kinds of ways in which genomes can be understood and manipulated.

The ability of "big data" tools to rapidly analyze large volume of data has made them particularly attractive. This is a story of how a particular set of tools (known as a sequence comparison or sequence matching or sequence mapping algorithms) has come to enjoy wide application and dominance in genomics. In the 1990s, these tools were critical for assembling genomes; in the 2000s they were necessary for making use of next-generation sequencing (NGS) data; and most recently they are central to the approaches that come under the label "precision medicine." Without these tools biologists could not have assembled genomes, nor used NGS data, nor begun to develop "precision medicine" approaches. These algorithms have also become exemplary for biologists as they deal with other large non-sequence-based "high-throughput" data sets. Of course, the kinds of big data tools described here are not the *only* methods deployed in genomics, but they do play a central role in some of the most visible new approaches to understanding organisms

in the genomic era. As such, analyzing these algorithms provides an important means through which we can gain insight into recent genomic practices.

Search algorithms promise to make sense of the Web by finding correlations and patterns amidst the trillions of bytes of words. Similarly, sequence comparisons hope to make sense of organisms and diseases by finding patterns amidst the trillions of bytes of data collected in high-throughput experiments. This connection is more than metaphorical: the algorithms described here show how the data of the Web and the data of genomes are subjected to the very same kinds of pattern-searching methods. By linking biological and "big data" practices, algorithms are playing a central role in reconstituting biomedical practice in ways that tie it to particular sorts of questions and problems. Many "big data" methods are designed for finding patterns in websites. But genomes and other 'omic data are not websites and are unlikely to be meaningful in the same way. Since we don't know, *a priori*, how genomes are organized or function, assuming that they are organized or function like texts necessarily constrains the possibilities for understanding them.

Scholars across a range of fields are increasingly paying attention to how software and data structures affect the world and our understanding of it (including Bowker & Star 1999; Bowker 2006; Manovich 1999; Manovich 2013; Kirschenbaum 2007). Material properties of information systems "constrain, shape, guide, and resist patterns of engagement and use" says Paul Dourish (2014). Brian Cantwell Smith argues that "the representational nature of computation implies something very strong: that it is not just the ontology of computation at stake; it is the nature of ontology itself." (Smith 1998, 42). In other words, how objects are represented in databases, algorithms, and data structures reveals much about what we consider objects to *be* in the world. These approaches suggest that algorithms can be used to

gain insight into ways of thinking and doing in specific fields. Because of the ways algorithms structure and order the biological world, examining algorithms can provide a means to understand the kinds of ways biologists think about and work with their objects of study. In other words, this essay takes algorithms as an analytic tool through which to interrogate how biologists know and work. Software is increasingly a mechanism through which practices, styles of thought, and ways of constituting and understanding the world move from domain to domain. In particular, algorithms carry with them not merely ways of working, but ways of organizing, categorizing, and valuing objects; this too, forms a critical part of software's contemporary power and significance.

Adrian Mackenzie has examined how bioinformatic practices have evolved as biologists had to deal with larger and larger amounts of sequence data, especially data emerging from NGS machines. This has involved paying increasing attention to the "logistics" (Mackenzie *et al* 2015) and multi-dimensionality (Mackenzie 2015) of sequence data. This essay seeks to build on Mackenzie's work by showing how large volumes of biological data require not only new modes of storage and new statistical approaches, but have also mobilized and developed particular kinds of algorithmic approaches.

Molecular biologists have long considered genes a sort of "book of life" (Kay 2000). However, sequence comparison algorithms have *operationalized* these textual metaphors, strengthening and deepening the associations between DNA sequence and text. Elsewhere I have described how the emergence of bioinformatics transformed biological practice, orienting it towards production, exchange, and circulation of nucleotide sequences (Stevens 2011b; Stevens 2013; see also Thacker 2005). Sequence comparison algorithms form a critical part of this transformation, allowing

DNA sequence to be manipulated, analyzed, and circulated in ways much like textual data. As such, bioinformatic modes of data-driven circulation contributed to and reinforced the notion that genomes could be appropriately and usefully rendered as texts. Kay argued the that "code" and "text" metaphors of molecular biology were ultimately misleading (particularly in over-emphasizing the role of DNA in controlling organisms). By strengthening the associations between DNA and text and embedding them in software, sequence comparison algorithms emphasize particular kinds of accounts and explanations of how organisms work. These accounts are based the assumption that biological data can meaningful in the same way as other kinds of human-generated texts.

In tracking these developments, I first examine "indexing" algorithms developed in the early 1990s and used to assemble the first complete draft of the human genome. Second, I analyze a set of algorithms designed to process data from NGS machines using the Burrows-Wheeler Transform. Data from NGS exhibited specific characteristics that made it susceptible to nascent "big data" methods. Finally, I turn to explicit attempts to apply "big data" tools to the solution of problems in genomic medicine. My analysis is based on published descriptions of the algorithms in the scientific literature as well as other accounts intended for pedagogical purposes or for the biotechnology industry. These particular algorithms are important because they comprise one significant way in which biologists seek to understand genomes and how they work; without sequence comparison and mapping, genomes could not be made legible and meaningful biological objects. Since algorithms themselves comprise ways of doing and making in genomics, the analysis of the ways in which these algorithms work provides an alternative method for gaining insight into biological practice.

I: The Application of Indexing Algorithms to Genomes

During the 1990s, "indexing" algorithms became critical tools for genomics. These algorithms provided the means through which biologists could computationally reconstruct whole genomes. This was achieved by applying techniques that had been developed for and applied to text-searching problems since the 1950s. By describing some of the history of indexing algorithms and showing how these methods were taken up in genomics, I show here how genomes came to be understood increasingly as texts. Genomes had to become "textual" in order to be assembled into meaningful biological objects.

One of the first effective (and now ubiquitous) "indexing" algorithms - the hash - was invented by Hans Peter Luhn at IBM in the 1950s (Knuth 1973, 540-541). In early 1953, Luhn wrote an internal IBM memo in which he suggested putting information into "buckets" in order speed up a search. Rather, than searching through a long list of items one by one, a computer search could be sped up by dividing up the list into "buckets" according to a simple procedure. For instance, if one had a long list of telephone numbers, they could be divided up according to the last digit of each number. This would create ten "buckets," each containing approximately one tenth of the full list. This is a simple hash index.

Luhn's thinking about such search problems was influenced by his thinking about searching texts. Since the 1940s, Luhn had been working on various ways of auto-indexing, auto-abstracting, and searching texts using machines. In 1947, Luhn invented a way of modifying a typewriter to make machine-readable documents using

magnetic marks, and during the 1950s Luhn developed a series of card-sorting machines designed for indexing an information retrieval (Belzer *et al* 1978, 139-151). Luhn's most important invention in this field was called KWIC: Key Words In Context. This procedure could quickly and automatically construct a kind of index for a set of titles. Each keyword appearing in the titles would appear alphabetically, "in context" (that is, showing the words appearing before and after it). Like Luhn's "bucket" system, KWIC relied on automatically re-arranging items in such a way that they could then be easily scanned or searched for the required information.

Although this now seems quite trivial, until Luhn's invention there was no practical way to quickly index a set of titles or documents. Extracting keywords was a painstaking process that required human eyes and brains. With the amount of information in science and business growing too fast for most people to keep up, KWIC was the 1950s equivalent of a search engine: it allowed users to rapidly locate the information they needed. KWIC resulted in the design and construction of hundreds of computerized indexing systems in the early 1960s including those used by the Chemical Abstracts Service, Biological Abstracts, and the Institute for Scientific Information. Luhn knew that his system was useful for business users too. In 1958, Luhn wrote an article for the *IBM Journal of Research and Development* called "A Business Intelligence System." Here, Luhn proposed a system that could "auto-abstract" documents, extract "action points," and distribute them to appropriate people within an organization (this is the basis of the field that became known as "Selective Dissemination of Information"; Luhn 1958).

Hashing functions are one important example of the powerful set of tools developed for text searching and indexing. Various types of hashes are now used in a wide range of applications including computer graphics, caches, cryptography,

telecommunications, and plagiarism and piracy detection. Indeed, the significance of indexing across all domains of computing has made it one of the most important domains of research in computer science. This research has led to the development of other indexing schemes including suffix trees (1973), suffix arrays (1989), the Burrows-Wheeler Transform (1994), and Ukkonen's algorithm (1995).

During the 1990s, indexing algorithms, including hash functions, became widely used tools in genomics. In the late 1980s, at the newly created National Center for Biotechnology Information at the National Institutes of Health, David Lipman oversaw the development of a new algorithm called the Basic Local Alignment Search Tool (BLAST; Altschul *et al* 1990). BLAST was initially designed for searching for a particular string of DNA text in a large DNA sequence database such as GenBank. This involved a sequence matching or sequence alignment problem: that is, finding the best match between one piece of DNA and another. BLAST works by identifying high-scoring "key words" of DNA (usually 11 letters for DNA), then creating an index of the locations of these words for searching, then trying to find matches between the target sequence and these words, and finally extending those matches outwards, producing longer and longer matching regions (for more details on these developments see Stevens 2011a).

This way of thinking about sequences a series of "words" had a significant impact on how biologists began to imagine sequence and how fragments of sequence related to one another. In particular, it made it possible to deploy different kinds of text-searching techniques on sequences and genomes. This opened up new possibilities for not only searching for words, but also assembling fragments into longer sentences and paragraphs. A single chromosome can be tens of millions of base pairs long. Sequencing technology in the 1990s allowed determining sequences of around five

hundred base pairs at a time. This presented a fundamental problem for genomics: how was it possible to join up all the five-hundred base pair fragments in the correct order to reconstruct the full genome? One important part of the strategy was to make many copies of a chromosome and then sequence lots of random overlapping fragments. In theory, the overlapping segments could then be matched with one another to reconstruct longer pieces.

This reconstruction of long sequences from overlapping sequence fragments is a problem of sequencing matching or alignment. Lining up two sequences end-to-end is simple. But assembling a whole chromosome would require finding the best matches between *tens of thousands* of such fragments. Here, the aim of sequence comparison was not locating sequences within databases, but rather the assembly of the sequence fragments produced by sequencing machines into whole chromosomes. One of the first algorithms to be successfully utilized for genome assembly on a large scale was the TIGR Assembler, developed at The Institute for Genome Research and used to assemble the sequence of *Haemophilus influenzae* in 1995 (the first free-living organism to be sequenced; Sutton *et al* 1995). TIGR's algorithm utilized a strategy similar to BLAST, identifying short matching "words" within the sequences as candidate overlapping regions and then creating a searchable hash index of such words.

The indexing step works like an index for a book – a full index would tell you the exact location of every occurrence of every word in a book. An index of a genome does the same thing, but instead of listing the words in alphabetical order, it uses clever ways of pointing to the locations of various "words," just as Luhn did with his "buckets," using a simple set of mathematical or logical operations. Algorithms like

the TIGR Assembler provided something like a KWIC for the genome, allowing rapid lookup.

Other indexing schemes such as suffix arrays and suffix trees have also been used extensively in genomics from the 1990s onwards. For example, suffix trees have been used for whole-genome alignment (Delcher *et al* 1999). Suffix arrays, first applied to biology by Udi Manber and Eugene Myers (1993) were adapted for applications including sequence alignment (Kielbasa *et al* 2011), error correction of sequences from genome sequencers (Ilie *et al* 2011), genome assembly (Gonella and Kurtz 2012), word counting (Kurtz *et al* 2008), and sequence clustering (Hazelhurst and Lipák 2011). One of the first textbooks for computational biology, *Algorithms on Trees, Strings, and Sequences* (Gusfield 1997) also emphasized suffix-tree approaches. Text searching using these techniques was critical to growing power of bioinformatics in the 1990s and early 2000s.

Significantly, the "competition" between the private company Celera Genomics and the publicly funded HGP was, in part, a competition over algorithms. The public project had previously considered and rejected the "whole genome shotgun" (WGS) method on the basis that it was not computationally possible to assemble a whole genome's worth of fragments. Celera, however, collaborated with Compaq to develop a high-speed supercomputing environment for this purpose. This system allowed Eric Anson and Eugene Myers to extend the TIGR Assembler into the basis for the Celera Assembler (Anson and Myers 1999; Myers *et al* 2000). This was used to sequence the *Drosophila* genome and eventually the human genome on a powerful computer cluster (Venter et al. 2001).

In 2000, Celera had bought the computer company Paracel for $283 million in order to acquire "the world's fastest sequence comparison supercomputer

(GeneMatcher™), high-throughput sequence analysis and annotation software tools, and a text search supercomputer (TextFinder™)" (Celera 2000). Paracel, spun out of the defense contractor TRW in 1992, primarily built hardware and software for intelligence-gathering; TextFinder and other software was designed to sift through communications for hidden patterns of letters and words; one of its main customers was the National Security Agency (Pollack 2000). Once purchased by Celera, this technology was re-deployed to searching for patterns in genomes. Here, as elsewhere, tool that had important and valuable applications in other contexts were rapidly appropriated and adapted for genomics.

Without these tools, DNA sequencing methods could only produce a jumble of random DNA fragments. Indexing became critical for biologists in imagining what a genome was, how it could be assembled, and how it worked; it showed one particular way in which the patterns of letters in DNA sequences could come to have meaning. The algorithms also allowed sets of tools designed for business and intelligence applications, especially those designed for manipulating texts, to be deployed in genomics. Genome sequences thus increasingly came to be organized, circulated, and analyzed like other forms of textual data. Unlike email messages, books, or scientific papers, however, DNA sequence is not written by humans. There is little reason to expect that the kinds of patterns found in human-generated texts would be found in genomes. Genomes have more recently begun to show themselves as more densely interconnected, more non-linear, and more complex than most human texts. Assuming that the patterns of genomes would be like patterns in texts necessarily closes off alternative possibilities for understanding how they work. Nevertheless, DNA sequences *were* analyzed as texts, reinforcing the notion that genomes *are* texts and that they could be meaningfully parsed as such.

II: Next-Generation Sequencing and the Burrows-Wheeler Transform

In the first decade of the 21st century, sequence comparison tools played a critical role in the analysis of so-called "next-generation" sequencing (NGS) data. Here too, sequence comparison algorithms framed the search for the causes of genetic disease in genomes as a problem of text searching. The massive volumes of NGS data could only be effectively used by once again borrowing and adapting powerful software designed for searching and manipulating texts.

Just after the conclusion of the HGP, several new DNA and RNA sequencing technologies became available to genomics. Many of these relied on techniques developed in the 1990s but took a decade or more to mature into commercial machines. In 2004, Roche Applied Sciences marketed their 454 FLX Pyrosequencer, in 2006 Illumina released the Solexa 1G Genetic Analyzer, and in 2007 Applied Biosystems began sales of the their SOLiD (Supported Oligonucleotide Ligation and Detection) machine (Stein 2008). These platforms allowed a substantial speeding-up of DNA and RNA sequencing. The first generation of Illumina's machines, for example, could produce around one gigabase of data (one third of one human genome's worth) in a "run" that could be completed in around a week. By 2011, newer machines could produce nearly one thousand times that much (one terabase of data) in a similar amount of time (Illumina 2013).

One important limitation of these NGS machines was the short "read-length." While the Sanger sequencing methods (used in the HGP) could reliably sequence from 500 to 1000 base pairs, early NGS machines could sequence only (randomly

selected) small chunks, often as few as 20-30 base pairs.[1] A one-gigabase run of an Illumina machine could produce around 30 million very short "reads" of DNA or RNA. This limited the *kinds* of work that could be done with these new technologies. Sequencing a new species (called *de novo* sequencing) was not possible since it would be impossible for sequence comparison algorithms to reconstruct so many short pieces into a full genome. Rather, NGS could be used to take "snapshots" of gene expression (RNA sequencing) or to identify mutations at specific sites on the genome.

The key to making NGS data useful was the ability to "map" each read from the sequencing machine to its specific location on the "reference" genome (that is, a genome that had been previously assembled based on Sanger sequencing).[2] This "mapping" step relied, once again, on sequence comparison algorithms to match the short reads to their correct places on the genome. The emergence of NGS, then, led to the development of a new set of specialized algorithms for mapping short reads.

Although some of the earliest "mapping" algorithms for NGS data (including ELAND, the Short Oligonucleotide Alignment Program (SOAP), SeqMap, and MAQ) used hashing and other indexing methods, the most successful and powerful of these (including Bowtie and SOAP2) deployed a new indexing method called the Burrows-Wheeler Transform (BWT).[3] The BWT was first described by Michael Burrows and David Wheeler (1994) and published by the DEC Systems Research Center in Palo Alto, California. BWT was designed as a compression algorithm; the main uses of BWT are in the "zip" software "bzip2" (http://www.bzip.org/). Data

1 In the 1990s most of the sequencing machines used to sequence the human genome were reliable up to about 500 base pairs. Later versions of Sanger sequencers were reliable closer to 1000 base pairs. Very early NGS machines had read lengths of 20-30 base pairs; common read lengths on current (2015) models are between 100 and 250 base pairs.
2 The move from Sanger sequencing to NGS can also be characterized as a move from constructing "reference genomes" to "references populations" characterized by their specific patterns of variations. On the production and use of "reference populations" see M'Charek (2005, 44-46).
3 For a review of these algorithms see Li and Homer 2010.

compression usually relies on finding repeated letters within strings of text. BWT involves a clever and reversible method of reordering strings alphabetically, maximizing the number of repeated letters and thereby increasing the efficiency of the compression.

Using the BWT for sequence matching and mapping takes advantage of some of the same special properties that make the algorithm useful for compression. In 2000, Paolo Ferragina and Giovanni Manzini (2000) discovered that the BWT of a long string of letters could be used as a space-efficient *index* of the original string. In other words, it could be used, like a hash table, as a way to look up occurrences of subsequences within the original string. The FM-index, as it was called, could be stored in a fraction of the space of an equivalent hash table (or suffix array; Langmead *et al* 2009). An index to a book tells us which important words or phrases occur on which pages of the book. A full index, listing every occurrence of every word and phrase in the book, would usually be much longer than the book itself.[4] The BWT-FM Index provides a way of generating a full index that is only two-thirds longer than the book itself. Significantly, though, BWT-FM looks and works nothing like a conventional index. It functions via a set of mathematical tricks that provide a convenient shortcut to the content.

For our purposes, there are two important points about the use of BWT in genomics. First, its use relies on treating sequence data as text – the algorithm relies on ordering and manipulating the characters in the string *lexicographically*. That is, it necessarily treats the DNA string as a written text. Second, BWT's ability to handle

4 The size of the index would depend on the content of the book and the minimum and maximum size of the words in the index. Indexing a genome is harder than indexing a book since there are no discrete words. For example, the words "wire door" would need to be indexed not only as "wire," "door" and "wire door," but also as "wi," "ir," "re," "ed," "do," "oo," "or," "wir," "ire," "red," "edo," "doo," "oor," "ired," "redo," and "edoo."

large quantities of textual data made it possible to find ways of utilizing the exponentially larger data volumes generated by NGS.

In particular, the BWT allowed biologists to use NGS data to address one specific problem. Beginning in 2005, genome-wide association studies (GWAS) were designed to identify the genomic sites responsible for particular human diseases or traits and to determine the "risk" factors associated with each site (Rose 2007, chapter 4). These studies involved hundreds or thousands of individuals, some possessing a particular trait (eg. autism or obesity) and some not. By examining hundreds of thousands (or even millions) or locations on the genomes of each individual using microarrays, biologists could use statistical techniques to highlight particular genomic mutations (SNPs) that were over-represented amongst those possessing the trait.

GWAS approaches, however, had limited success in explaining human diseases and phenotypic traits (Daly 2010). Attempts to apply GWAS to complex diseases (such as obesity or autism) correlated the diseases with hundreds or thousands or genomic loci (Visscher *et al* 2012b). Even apparently simple traits, such as height, appeared to be linked to hundreds of sites (Allen *et al* 2010). Even more problematic, even the contributions of this multitude of loci did not seem to be able to fully explain the heritability of these traits and diseases. This "missing heritability" problem seemed to suggest that something much more complex must have been going on between genomes and phenotypes (Manolio *et al* 2009).

GWAS advocates argued that the answer would lie in the collection of more data on human genomic variation (Visscher *et al* 2012a). Mapped NGS data could be used to exactly this. The use of NGS for GWAS-type studies, therefore, represented a doubling down on GWAS's data- and correlation-driven approach (Koboldt *et al* 2013). The basic idea was the same as GWAS, but now much more data could be

collected (in the form of the millions of reads from NGS mapped to the human genome). Moreover, no prior assumptions needed to be made about where to look for variation.[5] The genomic basis of human diseases and traits could be discovered, GWAS proponents believed, by searching more deeply for patterns of variation between genomes, using NGS.

Indeed, much of the interest in (and excitement about) NGS was based on its potential for extending GWAS methods by searching more deeply for variation. As such the most significant early uses of NGS technology were in performing genome-wide searches for human variation and the most critical challenges were in finding faster algorithms for performing alignment and mapping of short-reads (Zhang *et al* 2011). One influential early review of NGS (Shendure and Ji 2008) reported several uses for the new machines including "targeted discovery of mutations or polymorphisms," "mapping of structural rearrangements," "serial analysis of gene expression," and "large-scale analysis of DNA methylation"). All of these involved mapping reads to a reference genome in order to track patterns of variations. At a broad level, all of these experiments involved a "find" procedure – that is, searching for a specific pattern of text within a large set of textual fragments. While genome assembly required finding matching patterns within a single genome in order to discover overlaps, NGS approaches searched for matching patterns across multiple genomes in order to draw conclusions about variation between these genomes.

Sequence mapping allowed NGS to be directed towards asking and answering a specific kind of biological question. Namely, they allowed biologists to search genomes for patterns of variations that could be correlated with disease. NGS

5 GWAS had necessarily had to limit itself to looking for so-called "common variants." NGS could search for "rare variants."

algorithms were directed towards identifying over-represented patterns of DNA letters in large numbers of genomes. The massive volumes of short-read data from NGS machines could only be made useful and legible by deploying these specific kinds of powerful text-matching algorithms. The BWT was designed to take advantage of patterns in texts for the purposes of compression; in genomics, the algorithm could be deployed to rapidly search for textual patterns. Once again, the logic of human-generated *text* was applied to analyzing genes and genomes. The dominance of this "big data" approach has constrained the ways in which biologists have come to use NGS data and the kinds of meanings they expect to find within it.

III: Big Data Methods and Precision Medicine

Most recently, genomics has been touted as the means to achieve "precision medicine" – this involves the use of genomic and other data to tailor diagnostics and treatments for individuals. "Big data" tools are critical to this work. Again, however, these tools rely on adapting and re-deploying tools from other domains. Such tools are designed to address specific sorts of text-based problems and their use in genomics reflects this.

Between 1994 and 1997, Sergey Brin and Lawrence Page, while graduate students at Stanford, designed a new method for searching the expanding amount of information on the World Wide Web. This system relied on "crawling" the Web (downloading and storing pages) and creating an "index" that could be rapidly searched against a user query (Brin and Page 2000). A 1998 version of Brin and Page's system already had to download and index 25 million web pages. This required ways of managing large amounts of data that could be stored (and accessed)

across multiple computers. Brin and Page developed a method of creating "virtual files" – system they called "BigFiles" – that spanned multiple computers.

As the web grew rapidly in the early 2000s, keeping up with ways to store and index the growing volume of information was a priority for Google. Google continued to develop and refine its own file system for storing the vast quantities of information it required. This eventually became the Google File System, capable of managing files hundreds of gigabytes in size (Carr 2006). However, in addition to merely storing large volumes of information Google also needed to perform computations involving these huge data sets: summarizing of the numbers of pages crawled per host, or finding the set of the most frequent queries in a given day, for example (Dean and Ghemawat 2004).

In 2003, this requirement inspired the creation of Google's MapReduce software. The idea was that, for a large computation involving hundreds of computers, MapReduce would take care of all the details of distributing the calculation and aggregating the results. Engineers seeking to perform a task that required multiple machines could simply forget about this multiplicity and act as if the operation was going to be performed on a single computer - MapReduce would take care of all the details of dividing up the calculation. Its developers wrote: "we designed a new abstraction that allows us to express the simple computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data-distribution, and load balancing" (Dean and Ghemawat 2004).

Although Google kept the precise details of its software secret, it did publish enough information about MapReduce to allow others to implement similar systems. In 2004, software engineers Doug Cutting and Mike Cafarella began developing their own version, called Nutch. Yahoo, determined not to be outdone by Google, hired

Cutting and Cafarella to join their own search engine development team in 2005 (Metz 2011). Yahoo was also committed to keeping the project in the public domain, spinning off the project to a separate company called Hortonworks. The result was Hadoop (named after the elephant in the Dr. Seuss book *Horton Hears A Who*).

The kinds of problems for which MapReduce was designed and used suggest the kinds of problems that Google needed to solve. In the paper describing MapReduce, Dean and Ghemawat report on three kinds of tasks. First, they describe the performance of the software in performing a "grep" task, scanning through 10^{10} 100-byte records, searching for a relatively rare three-character pattern. Second, they describe how the algorithm executes a "sort" task, ordering 10^{10} 100-byte records. Third, the authors describe how MapReduce has been used to speed up Google's indexing of webpages (Dean and Ghemawat 2004, 10-11). Although the authors also note that MapReduce has found uses in other domains ("large-scale machine learning problems, clustering problems for Google News and Froogle products, extraction of data used to produce reports of popular queries (eg. Google Zeitgeist), extraction of properties of webpages for new experiments and products, and large-scale graph computations" (Dean and Ghemawat 2004)) many of these are essentially large-scale text-matching and pattern-finding problems. This sort of text-matching is an *exemplary* problem for Google and MapReduce. David Carr recounts a training assignment that would be typical for a new programmer hired by Google - using MapReduce to count all occurrences of words in a set of Web documents:

> In that case, the "map" would involve tallying all occurrences of each word on each page—not bothering to add them at this stage, just ticking off records for each one like hash marks on a sheet of scratch paper. The programmer would

> then write a reduce function to do the math—in this case, taking the scratch paper data, the intermediate results, and producing a count for the number of times each word occurs on each page (Carr 2006).

Although this is a simple problem in principle, the technique is the basis for a wide range of statistical-data mining problems at the center of Google's work. "[This sort of problem] is particularly key for Google, which invests heavily in a statistical style of computing, not just for search but for solving other problems such as automatic translation between human languages..." (Carr 2006).

Indeed, Google's attempts to "solve the world's problems" are based on exactly this type of approach. That is, they rely on efficient ways of counting occurrences and finding patterns in the occurrences of words in large data sets. Google Translate, Google's search, Google FluTrends, and Google's advertising systems all work in this way. Google Translate, for example, looks for statistical relationships between the ways words occur together across millions of human-translated documents (such as UN documents, which are translated into multiple languages). Google FluTrends examines patterns in search terms and associates them with particular locations. Google's Adsense attempts looks for patterns of words occurring together in order to indicate the "meaning" of webpages so as to make ads more relevant to the content alongside which they are placed (Levy 2011). For Google, solving problems means finding patterns in text.

In 2009, Michael Schatz, a scientist at the University of Maryland, began to apply Hadoop to biological problems (Hernandez 2013). Schatz used Hadoop to run genomics calculations on Amazon's EC2 "elastic" cloud computing service, finding

ways to drastically reduce the amount of time required to process his data. Like Google, sequence comparison algorithms seek patterns in large quantities of text. Schatz's contribution – a program he called Cloudburst – was to implement sequence mapping in a way that allowed it to be distributed and accelerated via Hadoop. "It is an accurate, fast, and cheap way of squeezing 1000 hours of computation into an afternoon..." Schatz's team reported (Bisciglia 2009).

The adaptation of Hadoop for biology is enabled by the fact that sequence comparison algorithms are conceptually similar to the kinds of problems MapReduce was designed to solve. Problems such as counting all occurrences of words in a (large) set of web documents are precisely analogous to counting "the number of occurrences of all length k substrings (k-mers) in a set of DNA sequences" (Schatz 2009). The problem that Schatz uses to illustrate the application of MapReduce to sequence analysis is exactly the kind of word-counting problem assigned to Google's novice programmers (Schatz 2009). In one sense, this is not surprising: the algorithm is simply being deployed for its original purpose. But since Hadoop is one of the few algorithms able to handle massive data volumes, this practically limits the kind of work and the kinds of "big data" problems biologists are able to tackle.

The use of Hadoop/MapReduce techniques in biology has made a range of new computational resources available for biological work (Taylor 2010). In particular, biology can now take advantage of "the cloud," doing its computing in Amazon's "elastic compute cloud" or elsewhere. The resources at companies such as Cloudera, designed for speeding online business operations, are now at the disposal of biology. Other companies, such as Spiral Genetics, DNAnexus, are now attempting to sell more specially adapted cloud services to biology labs (Hernandez 2013). These

developments also make biological work more dependent on the commercial infrastructures of "big data," especially cloud computing.

But more significantly, the ways in Hadoop has been used in biomedicine further suggest that this technology is suited to answering a very particular set of questions. The kinds of statistical methods all rely on discovering patterns and correlations within very large data sets. In effect, they once again extend the GWAS model and approach to much larger and more diverse data sets. A diagram from *GigaOm* shows the how this is supposed to work or how the cloud is (or will be) used to create "better medicine, brought to you by big data" (Harris 2012; see https://gigaom.com/2012/07/15/better-medicine-brought-to-you-by-big-data/). Tumour samples are collected from patients, sequenced, and stored in the cloud. But the key step is "map and match": the tumour genome is matched to a database of known tumours in order to determine "targeted drug therapy." This is a sequence comparison or text-matching problem. The aim is to identify patterns in the text of DNA that correlate with specific clinical outcomes. If one specific pattern of DNA words is found to be correlated with high survival rates across thousands of samples, and another pattern with the success of a particular treatment, then that information will be useful for characterizing and treating future cancers, proponents of this approach hope. Such "map and match" may be helpful for identifying and treating a particular patient's cancer, but that patient's data is then also collected and added to the database in order to refine future predictions. This is best described as a "find and count" problem – finding particular patterns of letters or words and counting their occurrence in order to identify which patterns are over-represented.

In another example, in June 2012, Google began a collaboration with the Institute for Systems Biology, adapting the Institute's "Regulome Explorer" software

for the Google Compute Engine. A "random forest" algorithm was put to work on data from the Cancer Genome Atlas to "explore associations between DNA, RNA, epigenetic, and clinical cancer data" (Thomas 2012). Here, once again, the aim is to find patterns in the text of DNA and RNA that correlate with specific clinical outcomes (such as survival rates). Finding out anything clinically useful about a specific patient or a specific tumor relies on data about a vast number of *other* patients and tumors. Individual results require massive aggregation. Removing the emphasis from the individual patient or the individualized disease leaves little room for recognizing exceptions or novelty – the individual's disease exists *only* in relation to the collective. This *de-individualization* of medicine means that, in order to benefit from "big data," all patients must subject themselves to these regimes of data collection and comparison.

"Big data" collaborations are premised on the idea that "big data" tools such as Hadoop will be able to process larger and larger quantities data fast enough to make a difference to patients (NextBio 2012; Brust 2012; Lohr 2015). Advocates of these approaches anticipate that novel insight will come not from new kinds of models or new approaches, but from scaling up. This kind of work does not rely on understanding the meaning or function of specific pieces of DNA (just as Google Translate does not understand the "meaning" of texts it translates). Rather, it relies on the notion that commonly occurring or over-represented patterns in the DNA (or RNA) have some functional or causal significance. It also relies, like Google's services, on having vast amounts of data at one's disposal – the more data, the better.

This is both logically and practically similar to the ways in which Google searches for and uses patterns in Web pages to establish their likely "meaning." The accuracy of Google Translate relies on comparing massive amounts of textual input,

and, as most users are aware, the more one uses Google's search, the better it gets at predicting what one want to know. Likewise, the efficacy of the cloud in medicine will rely consuming on massive amounts of patient data. It is only with tools such as Hadoop that biologists can hope to effectively mine these datasets. In this version of biomedicine, more data is always the key to finding the answer. Indeed, the more specific the result required, the more data will be needed.

The "precision medicine" paradigm relies on transforming not only sequence data, but also vast amounts of other kinds of biomedical and clinical data into text that can be searched for patterns and correlations. While the BWT rendered NGS data as text, Hadoop and other "big data" tools articular an even wider vision of biological work as a text-matching problem that involves pattern findings within sequence, clinical, environmental, and other forms of biomedical data. In other words, this approach is now no longer limited to DNA sequence data. These kinds of statistical methods all rely on discovering patterns and correlations within very large data sets. Hadoop can been applied to generate statistical associations between *any* kinds of data; all kinds of biological data can become "texts," to be analyzed in the same ways. This represents a narrowing, rather than an opening up, of the possibilities for thinking about the workings of genes, proteins, small molecules, and their mutual interactions.

Newer approaches such as environment-wide association studies (EWAS) and phenome-wide association studies (PheWAS) generate more and new types of data. But all of these are processed according to the same methods. In environment-wide association data, information about individuals' exposure to environmental toxins, diet, or physical activity is examined alongside their genomic and health data (Patel *et al* 2010). The goal is to correlate particular exposures or environmental factors with

the incident of specific diseases. In PheWAS, a single genomic location is compared across a range of individuals in order to identify the diseases or traits with which it is associated (Hebbring 2014). Although it is often described as an "alternative" or "complementary" approach to GWAS, it relies on the same method of searching for correlations and over-represented patterns using "big data" tools.

Conclusions

Although the application of Hadoop to biology is recent, I have argued that this is the latest stage of a longer borrowing of tools and from text-searching and text-matching problems. In the 1990s, indexing techniques borrowed from computer science (but originally developed for text matching and searching) became critical for assembling genomes. Later, the explosion of NGS data required even more powerful data-processing techniques. In particular, NGS data and analysis connected biomedicine to a variety of problems in software, databases, and the Web (including compression, indexing, and search). In genomics, the exemplary problem became – and has remained – finding patterns within very large sets of textual fragments.

Since the 1950s, computer scientists have devised powerful algorithms for searching, indexing, abstracting, and sorting text. Most recently, for search engines such as Google, matching and identifying patterns within the vast (hyper-)textual space of the World Wide Web is its central problem. As a result, computer scientists and Google's engineers have developed a set of extremely powerful solutions to this problem. The fixation of genomics on the "search" problem has much to do with the origins and usefulness of these methods in text searching and indexing, search engines, and pattern finding. "Search pervasively affects our view of the Internet and,

increasingly, of 'real life'" Frank Pasquale argues (Pasquale 2015, 59). Search pervades the way contemporary society thinks about our own lives, but has also influenced how many biologists think about "life itself." Genomics has capitalized on the powerful resources that the "search" problem has produced in framing its own problems and solutions. These textual tools have provided ready-made (or near ready-made) solutions to problems through adaptation of these tools for biology.

In the 2015 film *Ex Machina*, the anti-hero protagonist, Nathan, explains to one of his employees how he created an artificial intelligence. The key to making an intelligent robot was mining the world's Internet search data:

> Nathan: It was the weird thing about search engines. They were like striking oil in a world that hadn't invented internal combustion. They gave too much raw material. No one knew what to do with it… My competitors were fixated on sucking it up, and trying to monetize via shopping and social media. They thought engines were a map of what people were thinking. But actually, they were a map of *how* people were thinking. Impulse, response. Fluid, imperfect. Patterned, chaotic (Garland 2015).

This detail of Garland's film is suggestive of the value that the contemporary society places on search engines. The search engine has become a means for tracking trends, understanding language, and even understanding thought. "Google looks like the model for everything and the solution to every problem" argues Siva Vaidhyanathan (2011, 6). They have also become a powerful means, both practically and metaphorically, for understanding genomes.

What consequences does this problem-choice have for biology? Algorithms can reveal much about how groups of practitioners order their world and conceive of the objects in it. As Brian Cantwell Smith suggests, algorithms and data structures reveal how their designers and users organize categorize the world *outside* the computer too (Smith 1998). This suggests that "big data" approaches may be limiting our thinking about genomes and organisms. Conceiving biological systems in terms of text or pattern-matching limits the ways in which we might think about what they are and how they work. The notion that the genome is a meaningful text, a code to be broken, or a dataset with hidden patterns may close off other kinds of conceptions of biology.

But beyond this more speculative concern, the ubiquity of search poses other significant risks. First, such "search" methods may crowd out other ways of biological doing and knowing (eg. case studies) that do not deal in huge data volumes. High-throughput methods (such as next-generation sequencing) produce massive data volumes that seem to demand "big data" analysis that rely on pattern matching and correlation. Data production and data analysis become locked into a feedback cycle: more data demands faster analysis that justifies the production of yet more data. Although "big data" approaches actually amount to a very narrow set of text-matching tools, they are increasingly the only way of approaching problems involving large volumes of data.

A range of other high-throughput techniques now supplements NGS and machines such as flow cytometers, fMRIs, ChIP-seq, RNA-seq, and RIP-seq that also produce massive data sets (see, for instance, the variety of methods used in the ENCODE project; ENCODE at UCSC 2012). While these new techniques might, in principle, open up a wide range of new analytic possibilities, the very magnitude of

the data they produce seems to be limiting the kinds of approaches that can be imagined; "search" tools provide the only ready-made way for reducing and analyzing the large quantities of data they spew out. Increasing volumes and increasing messiness and diversity of data can be brought under control by returning to "big data" tools and techniques.[6] In a climate where bigger is better, the volume of data that can be produced becomes its own justification. The production of data, in turn, justifies the use of "big data" methods for finding patterns within it.

But many of these methods are tuned to finding patterns in websites. As disordered as the Web may be, it ultimately consists of (mostly) human writings that are (mostly) intended to be meaningful. There is no reason to expect that sequences will be meaningful in the same way. The limitations of GWAS suggest that genomes may function in ways that are more holistic, more densely interconnected, and more combinatoric than human-generated text. Applying text-specific algorithms to genomes begins with the *assumption* that biological data is like text, necessarily limiting the possibilities for discovering alternatives. Algorithms designed to find patterns will (almost) inevitably find them. The patterns that algorithms discover find may have more to do with *how* they are searching than *what* they are looking at. As with GWAS, however, more data may yield more correlations and more patterns, but they do not necessarily reveal which of these is likely to be biologically or clinically significant. More data may increase our ability to see patterns, but they do not tell us what those patterns mean (or what causes them).

Ultimately, Google's (or Facebook's) aim is to capture more of our attention and sell it to advertisers. As such, the company's focus in on aggregating user data in ways that can better respond to our needs, but also better predict our (consumer)

6 ENCODE has been criticized for its "big science" approach to biology (Eisen 2012).

behavior. Applied to biology, these technologies also seem to promise a "personalized" genomics and medicine. In both cases, the notion of "personalization" is an illusion; both rely on the massive aggregation of data from large numbers of people. What appears to be "personalized" is in fact dependent on millions of other individuals with whom you may share little or nothing in common.

In 2008, the direct-to-consumer personal genomics company, 23andMe began a service they called "23andWe." This part of their operation would aggregate their genomic customers data and correlate it with survey data (also from customers) that provided information about family and medical history, diet, and lifestyle information. This, 23andMe CEO Anne Wojcicki claimed, would allow the company to run it's own "experiments" – mining their database for linkages between genetic markers and diseases. This could even, Wojcicki went on, become an alternative to publicly-funded forms of biomedical research, speeding up the slow processes of peer review and approval for human subjects research (Wojcicki *et al* 2012).

In an impassioned critique of these developments, Sanford Kwinter argued that this amounted to nothing less than a "crypto-bio-prospecting" in which customers exchanged their bioproperty for a "service contract." The "logic of the startup, imposed on biology" resulted in the dangerous "handing over of biological endowment to a simplistic, discredited market model" (Wojcicki *et al* 2012). 23andWe's "experiments," and Kwinter's response, suggests how the "personalization" of medicine is tied to corporate ownership over data and the increasing empowerment of corporations within the frameworks of datafied medicine. Although 23andMe may represent an extreme example, other "big data" practices increasingly subject personal data to corporate control.

The deepening linkage, through algorithms, of biomedicine to commercial infrastructures (databases, clouds) indicates that society might come to see biomedicine in increasingly "market" terms. The rhetoric of "personal choice" and "individual responsibility" over "public health" already pervades biomedicine (Rose 2007, 124; Waldby 2006). The advent of "big data" methods increasingly makes medicine something to be sold. Users of Google and Facebook trade personal data (such as the contents of emails) for convenience. Critics of these platforms argue that ultimately the users get the worse end of this bargain – that the data collectors and aggregators have more to gain from massive amounts of aggregated personal data than can be used and sold (Vaidhyanathan 2011; Silverman 2015; Schneier 2015). Similarly in medicine, patients trade their genomic and health data for treatments. But, although beneficial to patients in the short term, this data may have more value to those able to collect, store, and analyze it in the long run. If the examples of personal data online are any indication, "big data" practices, by increasing the value of large, centralized data sets, may significantly contribute to the consumerization of medicine and the commodification of individuals that lies at the heart of "personalized" medicine.

References

Allen, H.L. *et al* (2010) Hundreds of Variants Clustered in Genomic Loci and Biological Pathways Affect Human Height. *Nature* 467, no 7321: 832-838.

Altschul, S.F. *et al* (1990) Basic local alignment search tool. *Journal of Molecular Biology* 215: 403-410.

Anson, E. and Myers, E. (1999) Algorithms for whole genome shotgun sequencing. In: Proceedings of RECOMB '99, Lyon, France, pp. 1-9.

Belzer, J. *et al*, eds. (1978) *Encyclopedia of Computer Science and Technology.* Vo1. 10. Linear and Matrix Algebra to Microorganisms. Marcel Dekker.

boyd, d. and Crawford, K. (2012) Critical questions for big data. *Information, Communication & Society* 15(5): 662-679.

Bisciglia, C. (2009) Analyzing human genomes with Apache Hadoop. Weblog, 15 October, Cloudera, http://blog.cloudera.com/blog/2009/10/analyzing-human-genomes-with-hadoop/, accessed 27 May 2015.

Bowker, G. and Star, S.L. (1999) *Sorting Things Out: Classification and its Consequences*. Cambridge: MIT Press.

Bowker, G. (2006) *Memory Practices in the Sciences.* Cambridge: MIT Press.

Brin, S. and Page, L. (2000) The anatomy of a large-scale hypertextual web search engine. Computer Science Department, Stanford University, http://infolab.stanford.edu/pub/papers/google.pdf, accessed 27 May 2015.

Brust, A. (2012) Cloudera and Mount Sinai: The structure of a big data revolution? *ZDNet*, 6 July, http://www.zdnet.com/article/cloudera-and-mount-sinai-the-structure-of-a-big-data-revolution/, accessed 27 May 2015.

Burrows, M. and Wheeler, D.J. (1994) A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-124.html, accessed 27 May 2015.

Carr, D.F. (2006) How Google Works: the Google File System. *Baseline*, 6 July, http://www.baselinemag.com/c/a/Infrastructure/How-Google-Works-1/4, accessed 27 May 2015.

Celera (2000) Celera Genomics to Acquire Paracel Inc. Press release, 20 March, https://www.celera.com/celera/pr_1056568938, accessed 18 September 2015.

Clarke, A.E. et al. (2003) Biomedicalization: Technoscientific Transformations of Health, Illness, and U.S. Biomedicine. *American Sociological Review* 68, no. 2: 161-194.

Dalton, C. and Thatcher, J. (2014) What Does a Critical Data Studies Look Like, and Why Do We Care? Seven Points for a Critical Approach to Big Data. *Society and Space*, http://societyandspace.com/material/commentaries/craig-dalton-and-jim-thatcher-what-does-a-critical-data-studies-look-like-and-why-do-we-care-seven-points-for-a-critical-approach-to-big-data/#comments, accessed 23 September 2015.

Daly, A.K. (2010) Genome-wide Association Studies in Pharmacogenomics. *Nature Reviews Genetics* 11: 241-246.

Dean, J. and Ghemawat, S. (2004) MapReduce: Simplified data processing on large clusters. Google Research Publications (appeared in OSDI '04: Sixth Symposium on Operating System Design and Implementation, San Francisco, California, December 2004),

http://static.googleusercontent.com/media/research.google.com/es/us/archive/mapreduce-osdi04.pdf, accessed 27 May 2015.

Delcher, A.L. *et al* (1999) Alignment of Whole Genomes. *Nucleic Acids Research* 27(11): 2369-76.

Dickson, S.P. *et al* (2010) Rare Variants Create Synthetic Genome-Wide Associations. *PLOS Biology,* 26 January, DOI: 10.1371/journal.pbio.1000294.

Dourish, P. (2014) No SQL: The Shifting Materialities of Database Technology *Computational Culture: A Journal of Software*, http://computationalculture.net/article/no-sql-the-shifting-materialities-of-database-technology, accessed 18 September 2015.

Eisen, M. (2012) Blinded by Big Science. Weblog entry, 10 September, www.michaeleisen.org/blog/?p=1179, accessed 23 September 2015.

ENCODE at UCSC (2012) ENCODE Experiment Matrix, http://genome.ucsc.edu/ENCODE/dataMatrix/encodeDataMatrixHuman.html, accessed 27 May 2015.

Ferragina, P. and Manzini, G. (2000) Opportunistic data structures with applications. Foundations of Computer Science. Proceedings, 41st Annual Symposium, pp. 390-398. IEEE, 2000.

García-Sancho, M. (2012) *Biology, Computing, and the History Molecular Sequencing: From Proteins to DNA, 1945-2000.* Palgrave-Macmillan.

Garland, A. (2015) *Ex Machina* (film). Writer and director: Alex Garland.

Gitelman, L., ed. (2013) *Raw Data is an Oxymoron*. Cambridge: MIT Press.

Gonella, G and Kurtz, S. (2012) Readjoiner: A Fast and Memory Efficient String Graph-based Sequence Assembler. *BMC Bioinformatics* 13(1): 1-19.

Griffin, A.M. and Griffin, H.G. (1994) *Computer Analysis of Sequence Data, Part I*. Totowa, NJ: Humana Press.

Gusfield, D. (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Harris, D. (2012) Better medicine, brought to you by big data. *GigaOm*, 15 July, https://gigaom.com/2012/07/15/better-medicine-brought-to-you-by-big-data/, accessed 27 May 2015.

Hazelhurst, S. and Lipák, Z. (2011). KABOOM! A New Suffix Array Based Algorithm For Clustering Expression Data. *Bioinformatics* 27(24): 3348-55.

Hebbring, S.J. (2014) The Challenges, Advantages and Future of Phenome-Wide Association Studies. Immunology 141(2): 157-65.

Helland, P. (2011) If you have too much data, then "good enough" is good enough. *ACM Queue*, 23 May, http://queue.acm.org/detail.cfm?id=1988603, accessed 27 May 2015.

Hernandez, D. (2013) Data crunchers ditch Hadoop for homegrown software. *Wired*, 20 February, http://www.wired.com/2013/02/genetic-data-glut/, accessed 27 May 2015.

Ilie, L. *et al* (2011) HiTEC: Accurate Error Correction in High-Throughput Sequencing Data. *Bioinformatics* 27(3): 295-302.

Illumina (2013) An introduction to next-generation sequencing technology, http://res.illumina.com/documents/products/illumina_sequencing_introduction.pdf, accessed 27 May 2015.

Kay, L.E. (2000) *Who Wrote the Book of Life? A History of the Genetic Code*. Stanford University Press.

Keller, E.F. (2015) The Postgenomic Genome. In: S. Richardson and H. Stevens (eds.) *Postgenomics: Perspectives on Biology After the Genome*. Durham and London: Duke University Press, pp. 9-31.

Kielbasa, S.M. *et al* (2011) Adaptive Seeds Tame Genomic Sequence Comparison. *Genome Research* 21: 487-93.

Kirschenbaum, M. (2007) *Mechanisms: New Media and the Forensic Imagination*. Cambridge, MA: MIT Press.

Kitchin, R. (2014) *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*. SAGE Publications.

Knuth, D.E. (1973) *The Art of Computer Programming*, Volume 3, "Sorting and Searching." Addison-Wesley.

Koboldt, D.C. *et al* (2013) The Next-Generation Sequencing Revolution and its Impact on Genomics. *Cell* 155(1): 27-38.

Kurtz, S. *et al* (2008) A New Method to Computer k-mer Frequencies and its Application to Annotate Large Plant Genomes. *BMC Genomics* 9(1): 1-18.

Langmead, B. *et al* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10: R25.

Leinonen, R. et al (2011) The sequence read archive. *Nucleic Acids Research* 39: D19-D21.

Levy, S. (2011) *In the Plex: How Google Thinks, Works, and Shapes Our Lives*. Simon & Schuster.

Li, H. and Homer, N. (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics* 11(5): 473-483.

Lipman, D.J. and Pearson, W.R. (1985) Rapid and sensitive protein similarity searches. *Science* 227 (4693): 1435-1441.

Lohr, S. (2015) On the case at Mount Sinai, It's Dr. Data. *New York Times*, 7 March, BU1.

Luhn, H.P. (1958) A business intelligence system. *IBM Journal of Research and Development* 2(4): 314.

MacClellan, J. and King, M.C. (2010) Genetic Heterogeneity in Human Disease. *Cell* 141: 201-217.

Mackenzie, A. *et al* (2015) Post-archival Genomics and the Bulk Logistics of DNA Sequences. *Biosocieties* 11(1): 82-105.

Mackenzie, A. (2012) More Parts than Elements: How Databases Multiply. *Environment and Planning D: Society and Space* 30: 335-350.

Mackenzie, A. (2015b) Machine learning and genomic dimensionality. In: S. Richardson and H. Stevens (eds.) *Postgenomics: Perspectives on Biology After the Genome*. Durham and London: Duke University Press, pp. 73-102.

Manber, U. and Myers, E. (1990) Suffix arrays: a new method of on-line string searches. In: Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 319-327.

Manolio, T.A. *et al* (2009) Finding the Missing Heritability of Complex Diseases. *Nature* 461, no. 7265: 747-753.

Manovich, L. (1999) Database as a Symbolic Form. *Millennium Film Journal* 34 (Fall).

Manovich, L. (2014) *Software Takes Command*. Bloomsbury Academic.

Mayer-Schönberger, V. and Cukier, K. (2013) *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Houghton Mifflin Harcourt.

Metz, C. (2011) How Yahoo spawned Hadoop, the future of big data. *Wired*, 18 October, http://www.wired.com/2011/10/how-yahoo-spawned-hadoop/, accessed 27 May 2015.

Myers, E. *et al* (2000) Whole-genome assembly of Drosophila. *Science* 287: 2196-2204.

Nature (2011) Best is Yet to Come. *Nature* 470 (10 February): 140.

NextBio (2012) NextBio and Intel collaborate to optimize the Hadoop stack and advance big data technologies in genomics, Press release, 11 July, http://www.nextbio.com/b/corp/pressReleases.nb#pr40, accessed 27 May 2015.

Pasquale, F. (2015) *The Black Box Society: The Secret Algorithms That Control Money and Information*. Cambridge and London: Harvard University Press.

Patel, C.J. *et al* (2010) An Enviroment-Wide Association Study (EWAS) on Type 2 Diabetes Mellitus. *PLoS One* DOI: 10.1371/journal.pone.0010746.

Pollack, A. (2000) Technology; Supercomputers Track Human Genome. *New York Times*, 28 August.

Rose, N. (2007) *The Politics of Life Itself: Biomedicine, Power, and Subjectivity in the Twenty-First Century*. Princeton: Princeton University Press.

Ruppert, E. *et al* (2015) Socializing Big Data: From Concept to Practice. CRESC Working Paper No. 138, The University of Manchester and Open University.

Sanger, F.S. *et al* (1977) DNA sequencing with chain-termination inhibitors. *Proceedings of the National Academy of Sciences USA* 74(12): 5463-5467.

Schatz, M. (2009) Cloudburst: Highly sensitive read mapping with MapReduce. *Bioinformatics* 25(11): 1363-1369.

Schneier, B. (2015) *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World*. New York: Norton.

Science (2001) Epigenetics. *Science*, special issue, 293, no. 5532: 1001-1208.

Shendure, J. and Ji, H. (2008) Next-Generation DNA Sequencing. *Nature Biotechnology* 26: 1135-45.

Shumway, M. *et al* (2010) Archiving next-generation sequencing data. *Nucleic Acids Research* 38: D870-871.

Silverman, J. (2015) *Terms of Service: Social Media and the Price of Constant Connection*. New York: Harper.

Smith, B.C. (1998) *On the Origin of Objects*. MIT Press.

Stein, R. A. (2008) Next-generation sequencing update. *Genetic Engineering & Biotechnology News* 28(15), 1 September, http://www.genengnews.com/gen-articles/next-generation-sequencing-update/2584/, accessed 27 May 2015.

Stevens, H. (2011a) Coding Sequences: A History of Sequence Comparison Algorithms as a Scientific Instrument. *Perspectives on Science* 19(3): 263-299.

Stevens, H. (2011b) On the Means of Bioproduction: Bioinformatics and How to Make Knowledge in a High-Throughput Genomics Laboratory. *Biosocieties* 6(2): 217-242.

Stevens, H. (2013) *Life Out of Sequence: A Data-Driven History of Bioinformatics*. Chicago: University of Chicago Press.

Stevens, H. (2015) Networks: Representations and Tools in Postgenomics. In: S. Richardson and H. Stevens (eds.) *Postgenomics: Perspectives on Biology After the Genome*. Durham and London: Duke University Press, pp. 103-125.

Sutton *et al* (1995) TIGR Assembler: a new tool for assembling large shotgun sequencing projects. *Genome Science & Technology* 1(1): 9-19.

Taylor, R.C. (2010) An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC Bioinformatics* 11(Suppl 12): S1.

Thacker, E. (2005) *The Global Genome: Biotechnology, Politics, and Culture*. Cambridge: MIT Press.

Thomas, U.G. (2012) Google works with ISB to evaluate life sciences as application area for new cloud infrastructure. *Genomeweb*, 20 July, https://www.genomeweb.com/informatics/google-works-isb-evaluate-life-sciences-application-area-new-cloud-infrastructur, accessed 27 May 2015.

Turnbaugh, P.J. *et al* (2007) Feature: The Human Microbiome Project. *Nature* 449, no. 7164: 804-810.

Vaidhyanathan, S. (2011) *The Googlization of Everything (And Why We Should Worry)*. Berkeley: University of California Press.

Venter, J.C. *et al* (2001) The Sequence of the Human Genome. *Science* 291, no. 5507: 1304-1351.

Venter, J.C. *et al* (2004) Environmental Shotgun Sequencing of the Sargasso Sea. *Science* 304, no. 5667: 66-74.

Visscher, P.M. *et al* (2012a) Evidence-Based Psychiatric Genetics, AKA the False Dichotomy Between the Common and Rare Variant Hypotheses. *Molecular Psychiatry* 17, no. 5: 474-485.

Visscher, P.M. *et al* (2012b) Five Years of GWAS Discovery. *American Journal of Human Genetics* 90, no. 1: 7-24.

Wojcicki, A. *et al* (2012) Deleterious Me: Whole Genome Sequencing, 23andMe, and the Crowd-Sourced Health Care Revolution. Science and Democracy Lecture Series, Harvard Kennedy School, 18 April. Available at: https://vimeo.com/40657814

Zhang, J. *et al* (2011) The Impact of Next-Generation Sequencing on Genomics. *Journal of Genetics and Genomics* 38(3): 95-109.