Published in final edited form as:

Commun ACM. 2016 August ; 59(8): 72–80. doi:10.1145/2957324.

Computational Biology in the 21st Century: Scaling with Compressive Algorithms

Bonnie Berger [professor in CSAIL],
Department of Mathematics and EECS at Massachusetts Institute of Technology, Cambridge, MA.

Noah M. Daniels [postdoctoral associate in CSAIL], and
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA.

Y. William Yu [graduate student in CSAIL]
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA.

Computational biologists answer biological and biomedical questions by using computation in support of—or in place of—laboratory procedures, hoping to obtain more accurate answers at a greatly reduced cost. The past two decades have seen unprecedented technological progress with regard to generating biological data; next-generation sequencing, mass spectrometry, microarrays, cryo-electron microscopy, and other high-throughput approaches have led to an explosion of data. However, this explosion is a mixed blessing. On the one hand, the scale and scope of data should allow new insights into genetic and infectious diseases, cancer, basic biology, and even human migration patterns. On the other hand, researchers are generating datasets so massive that it has become difficult to analyze them to discover patterns that give clues to the underlying biological processes.

Certainly, computers are getting faster and more economical; the amount of processing available per dollar of computer hardware is more or less doubling every year or two; a similar claim can be made about storage capacity (Figure 1).

In 2002, when the first human genome was sequenced, the growth in computing power was still matching the growth rate of genomic data. However, the sequencing technology used for the Human Genome Project—Sanger sequencing—was supplanted around 2004, with the advent of what is now known as next-generation sequencing. The material costs to sequence a genome have plummeted in the past decade, to the point where a whole human genome can be sequenced for less than US$1,000. As a result, the amount of genomic data available to researchers is increasing by a factor of 10 every year.

This growth in data poses significant challenges for researchers.[25] Currently, many biological "omics" applications require us to store, access, and analyze large libraries of data. One approach to solving these challenges is to embrace cloud computing. Google, Inc. and the Broad Institute have collaborated to bring the GATK (Genome Analysis Toolkit) to the Google cloud (https://cloud.google.com/genomics/gatk). Amazon Web Services are also commonly used for computational biology research and enterprise (for example, DNAnexus).[31] However, while cloud computing frees researchers from maintaining their own datacenters and provides cost-saving benefits when computing resources are not needed continuously, it is no panacea. First and foremost, the computer systems that make up those cloud datacenters are themselves bound by improvements in semiconductor technology and Moore's Law. Thus, cloud computing does not truly address the problem posed by the faster-than-Moore's-Law exponential growth in omics data. Moreover, in the face of disease outbreaks such as the 2014 Ebola virus epidemic in West Africa, analysis resources are needed at often-remote field sites. While it is now possible to bring sequencing equipment and limited computing resources to remote sites, Internet connectivity is still highly constrained; accessing cloud resources for analytics may not be possible.

Computer scientists routinely exploit the structure of various data in order to reduce time or space complexity. In computational biology, this approach has implicitly served researchers well. Now-classical approaches such as principal component analysis (PCA) reduce the dimensionality of data in order to simplify analysis and uncover salient features.[3] As another example, clever indexing techniques such as the Burrows-Wheeler Transform (BWT) take advantage of aspects of sequence structure[3] to speed up computation and save storage. This article focuses on cutting-edge algorithmic advances for dealing with the growth in biological data by explicitly taking advantage of its unique structure; algorithms for gaining novel biological insights are not its focus.

Types of Biological Data

In the central dogma of molecular biology, DNA is transcribed into RNA, which is translated by the ribosome into polypeptide chains, sequences of amino acids, which singly or in complexes are known as proteins. Proteins fold into sophisticated, low-energy structures, which function as cellular machines; the DNA sequence determines the amino acid sequence, which in turn determines the folded structure of a protein. This structure ultimately determines a protein's function within the cell. Certain kinds of RNA also function as cellular machines. Methods have been developed to gather biological data from every level of this process, resulting in a massive influx of data on sequence, abundance, structure, function, and interaction of DNA, RNA, and proteins. Much of this data is amenable to standard Big Data analysis methods; however, in this article we focus on examples of biological data that exhibit additional exploitable structure for creating scalable algorithms.

Sequence data, either nucleotide sequences (using a four-letter alphabet representing the four DNA or RNA bases) or protein sequences (using a 20-letter alphabet representing the 20 standard amino acids) are obtained in several ways. For both protein and RNA sequence data, mass spectrometry, which can determine protein sequence and interactions and RNA-seq, which can determine RNA sequence and abundance allow scientists to also infer the expression of the gene to which it might translate play central roles. However, with the advent of next-generation sequencing (NGS) technologies, the greatest volume of sequence data available is that of DNA. To better understand the structure of NGS sequence data, we will expand on NGS methodologies.

At the dawn of the genomic era, Sanger sequencing was the most widely used method for reading a genome. More recently, however, NGS approaches, beginning with Illumina's "sequencing by synthesis," have enabled vastly greater throughput due to massive parallelism, low cost, and simple sample preparation. Illumina sequencing and other NGS approaches such as SOLiD, Ion Torrent, and 454 pyrosequencing do not read a single DNA molecule end-to-end as one could read through a bound book. Instead, in *shotgun sequencing*, DNA molecules are chopped into many small fragments; from these fragments we generate *reads* from one or both ends (Figure 2a). These reads must be put together in the correct order to piece together an entire genome. Current reads typically range from 50 to 200 bases long, though longer reads are available with some technologies (for example, PacBio). Because no sequencing technology is completely infallible, sequencing machines also provide a *quality score* (or measure of the confidence in the DNA base called) associated with each position. Thus, an NGS read is a string of DNA letters, coupled with a string of ASCII characters that encode the quality of the base call. A sequencing run will produce many overlapping reads.

While measuring abundance to generate gene expression data (for more information, see the Source Material that accompanies this article in the ACM Digital Library) lends itself to cluster analysis and probabilistic approaches, the high dimensionality and noise in the data present significant challenges. Principal Component Analysis has shown promise in reducing the dimensionality of gene expression data. Such data and its challenges have been the focus of other articles,[3] and thus will be only lightly touched upon here.

As mentioned earlier, function follows form, so in addition to sequence and expression, structure plays an important role in biological data science. However, we are not interested in only RNA and protein structures; small chemical compounds represent an additional source of relevant structural data, as they often interact with their larger RNA and protein brethren. Physical structures of molecules can be determined by X-ray crystallography, NMR, electron microscopy, and other techniques. Once determined, there are a variety of ways of representing these structures, from labeled graphs of molecular bonds to summaries of protein domains. These representations can then be stored in databases such as Pub-Chem or the Protein Data Bank, and are often searched through, for example, for potential small molecule agonists for protein targets. Importantly, as we will expand upon later, interesting biomolecules tend to be sparse and non-randomly distributed in many representational spaces, which can be used for accelerating the aforementioned searches.

When examining more complex phenomena than single proteins or compounds, we often look to synthesize things together into a systems-level understanding of biology. To that end, we frequently use networks to represent biological data, such as the genetic and physical interactions among proteins, as well as those in metabolic pathways.[3] While standard network science tools have been employed in these analyses—for example, several approaches make use of diffusion or random walks to explore the topology of networks[9,11]—they are often paired with more specific biological data, as seen in IsoRank[32] and IsoRankN's[21] use of conserved biological function in addition to random walks for global multiple network alignment. Other tools solve other biological problems, such as MONGOOSE,[10] which analyzes metabolic networks. However, given its breadth, biological network science is beyond the scope of this article.

Challenges with Biological Data

Given DNA or RNA reads from NGS technologies, the first task is to *assemble* those fragments of sequence into contiguous sequences. The assembly problem is analogous to the problem of reconstructing a book with all its pages torn out. *De novo* assembly is beyond the scope of this article, but is possible because the sequence is covered by many overlapping reads;[3] for this task, the *de Bruijn graph* data structure is commonly used.[6] Often, however, a reference genome (or in the case of RNA, *transcriptome*) is available for the organism being sequenced; the establishment of a human reference genome was indeed the purpose of the Human Genome Project.

When a reference sequence is available, NGS reads can be mapped onto this reference (Figure 2c). Continuing the book analogy, it is much easier to reconstruct a book with all its pages torn out when one has another (perhaps imperfect) copy of that book to match pages to. Mapping allows the differences between the newly sequenced genome and the reference to be analyzed; these differences, or variants, may include single-nucleotide polymorphisms (SNPs, which are the genetic analogue to bit-flips, see Figure 2b), insertions or deletions, or larger-scale changes in the genome. Determining the differences between an individual genome and a reference is known as *variant calling*. While reference-based read mapping is a fundamentally simpler problem than de novo assembly, it is still computationally complex, as gigabytes or terabytes of reads must each be mapped onto the reference genome, which can range from millions (for bacteria) to billions (for mammals) of base pairs. As an example, the ICGC-TCGA Pan Cancer Analysis of Whole Genomes (PCAWG)[36] brings together more than 500 top cancer researchers from about 80 institutions in a coordinated manner with the goal of mapping the entire mutational landscape of 37 common cancer types. Currently, each sample requires seven hours to download even on an institutional connection. Importantly, researchers do not trust the provided mapping, and thus they redo mappings. The time spent on mapping is about 50% of the overall time spent on the sequence analysis pipeline. As read mapping is typically the most costly step in NGS analysis pipelines (for example, GATK), any improvement to existing mappers will immediately accelerate sequence analysis studies on large read datasets.

Driven by the plummeting costs of next-generation sequencing, the 1000 Genomes Project[1] is pursuing a broad catalog of human variation; instead of producing a single reference genome for a species, many complete genomes are catalogued. Likewise, WormBase and FlyBase are cataloguing many different species and strains of the *Caenorhabditis* worm and *Drosophila* fruit fly, respectively. These genomes are enabling cross-species inference, for example about genes and regulatory regions, and thus insights into function and evolution.[3] Again, the sheer enormity of sequencing data is problematic for storage, access, and analysis.

Given a sequenced genome, the next natural questions ask what genes (genomic regions that code for proteins) are present, what structure each resulting protein takes, and what biological function it performs. Identifying likely genes is a well-studied problem[3] beyond the scope of this article. However, determining evolutionary relationships, structure, and function is at the heart of current research in computational biology. Since some organisms (known as *model organisms*) are better studied than others, and evolution is known to conserve sequence, structure, and function, a powerful approach to determine these attributes is to search for similar sequences about which more is known. This so-called *homology search* entails searching for approximate matches in databases of known gene or protein sequences. The homology search problem was believed to be solved previously; Basic Local Alignment Search Tool (BLAST)[3] has been the standard tool for performing homology (similarity) search on databases of nucleotide and protein sequences. BLAST takes a “seed-and-extend” approach; it looks for small, *k*-mer matches that might lead to longer matches, and greedily extends them, ultimately producing a sequence alignment between a query and each potential database hit. However, BLAST’s running time scales linearly with the size of the database being searched, which is problematic as sequence databases continue to grow at a faster rate than Moore’s Law.

On a potentially even larger scale is the growth of *metagenomic* data. Metagenomics is the study of the many genomes (bacterial, fungal, and even viral) that make up a particular environment. Such an environment could be soil from a particular region (which can lead to the discovery of new antibiotics[14]), or it could be the human gut, whose microbiome has been linked to human-health concerns including Autism Spectrum Disorder,[23] Crohn’s Disease, and obesity.

Metagenomics fundamentally asks what organisms are present, and, in the case of a microbiome such as the gut, what metabolic functions it can accomplish as a whole. One way of addressing this problem is to attempt to map NGS reads from a metagenomic sample onto a set of reference genomes that are expected to be present. This is exactly the read-mapping problem discussed early, but with many reference genomes, compounding the computational requirements. A second way is to perform homology search on a protein sequence database; exact or nearly exact matches imply the presence of a species, while more distant hits may still give clues to function. For this task, BLASTX[2] is commonly used to translate nucleotide reads into their possible protein sequences, and search for them in a protein database. The difficulty is the datasets required to shine any light on these questions, namely from “shotgun” metagenomics, are gigantic and vastly more complex than standard genomic datasets. The massive data results in major identification challenges for certain bacterial, as well as viral, species, and genera.[19]

The computational study of drugs and their targets based on chemical structure and function is known as *chemogenomics*.[5] In the fields of drug discovery and drug repurposing, the prediction of biologically active compounds is an important task. Computational high-throughput screening eliminates many compounds from laborious wet-lab consideration, but even computational screening can be time consuming.

Chemogenomics typically relies on comparing chemical graph structures to identify similar molecules and binding sites. Furthermore, comparing chemical graph structures typically involves computing the maximal common subgraph (MCS), an NP-hard problem. However, there are an increasing number of such chemical compounds to search; the NCBI’s Pub-Chem database has grown from 31 million compounds in January 2011 to 68 million in July 2015.

The continued ability to store, search, and analyze these growing datasets hinges on clever algorithms that take advantage of the structure of, and redundancy present in, the data. Indeed, these growing datasets "threaten to make the arising problems computationally infeasible."[3]

State-of-the-Art Approaches to Meet These Challenges

Techniques for reference-based read mapping typically rely on algorithmic approaches such as the Burrows-Wheeler transform (BWT), which provides efficient string compression through a reversible transformation, while the FM-index data structure is a compressed substring index, based on the BWT, which provides efficient storage as well as fast search.[3] BWA (Burrows-Wheeler Aligner) uses the BWT, while the Bowtie[2] mapper further relies on the FM-index for efficient mapping of NGS reads.[3] The Genome Multitool (GEM) mapper[24] also uses an FM-index coupled with dynamic programming in a compressed representation of the reference genome, in order to prune the search space when mapping reads to a reference genome. Masai[33] and mrsFAST[15] use an "approximate seed" approach to index the space of possible matches, likewise pruning the search space; however, the bulk of its runtime is spent on the extend phase. State-of-the-art mapper mrsFAST-Ultra achieves improvements in efficiency based on machine architecture rather than leveraging redundancy in the data itself with near-perfect sensitivity, but only for the case where there are no insertions and deletions (indels).[16] Even with these approaches, read mapping remains a significant bottleneck in genomic research.[3]

Compressing reads for storage is necessary should researchers wish to apply more advanced mapping tools or other analysis in the future.[4] As stated earlier, NGS reads consist of a sequence string and associated quality scores, the latter of which generally uses more space when compressed. By taking advantage of biological structure, both parts of NGS reads can be better compressed. Unlike some other approaches to compressing quality scores in the literature,[4,26] Quartz[39] takes advantage of the fact that midsize *l*-mers can in many cases almost uniquely identify locations in the genome, bounding the likelihood that a quality score is informative and allowing for lossy compression of uninformative scores. Because Quartz's lossy compression injects information from the distribution of *l*-mers in the target genome, it demonstrates not only improved compression over competing approaches, but slightly improves the accuracy of downstream variant-calling.[39] Similarly, for the sequence component of the read, Mince[27] takes advantage of sequence redundancy by grouping similar reads (those that share a common short—15bp—substring) together into buckets, allowing that common substring to be removed and treated as the bucket label, so that each read in the compressed representation comprises only its unique differences from the bucket label. This approach allows a general-purpose compressor to achieve better compression. SCALCE[3] also relies on a "boosting" scheme, reordering reads in such a way that a general-purpose compressor achieves improved compression.

Recent advances in metagenomic search tools have relied on two improvements over BLASTX: indexing and alphabet reduction. RapSearch2[40] relies on alphabet reduction and a collision-free hash table. The alphabet reduction, as it is reversible, can be thought of as a form of lossless compression; a 20-letter amino acid alphabet is mapped onto a smaller alphabet, with offsets stored to recover the original sequence in the full alphabet. The hash table provides an efficient index of the database to be searched. DIAMOND[7] also relies on alphabet reduction, but uses "shaped seeds"—essentially, *k*-mers of length 15–24 with wildcards at 9–12 specific positions—instead of simple *k*-mer seeds to index the database. DIAMOND demonstrates search performance three to four orders of magnitude faster than BLASTX, but still linear in the size of the database being searched.

Recent work on gene expression has explored additional ways to exploit the high-dimensional structure of the data. SPARCLE (SPArse ReCovery of Linear combinations of Expression)[28] brings ideas from compressed sensing[8] to gene expression analysis. Another recent and novel approach to exploiting the structure of gene expression space is Parti (Pareto task inference),[17] which describes a set of data as a polytope, and infers the specific tasks represented by vertices of that polytope from the features most highly enriched at those vertices.

The most widely used chemogenomics search is the Small Molecule Sub-graph Detector (SMSD),[29] which applies one of several MCS algorithms based on the size and complexity of the graphs in question. Notably, large chemical compound databases, such as PubCHEM, cannot be searched on a laptop computer with current tools such as SMSD.

Structure of Biological Data

Fortunately, biological data has unique structure, which we later take advantage of to perform search that scales sublinearly in the size of the database.[38] The first critical observation is that much biological data is highly redundant; if a computation is performed on one human genome, and a researcher wishes to perform the same computation on another human genome, most of the work has already been done.[22] When dealing with redundant data, clustering comes to mind. While cluster-based search is well studied,[20] conventional wisdom holds that it provides a constant factor speed-up over exhaustive search.

Beyond redundancy, however, another attribute of large biological datasets stands out. Far fewer biological sequences exist than could be enumerated, but even more so, those that exist tend to be highly similar to many others. Thanks to evolution, only those genes that exhibit useful biological function survive, and most random sequences of amino acids would not be expected to form stable structures. Since two human genomes differ on average by only 0.1%, a collection of 1,000 human genomes contains less than twice the unique information of a single genome.[22] Thus, not only does biological data exhibit redundancy, it also tends not to inhabit anywhere near the entire feasible space (Figure 3). It seems that physical laws—in this case, evolution—constrain the data to a particular subspace of the Cartesian space.

One key insight related to redundancy is that such datasets exhibit low *metric entropy*.[38] That is, for a given cluster radius r_c and a database D, the number k of clusters needed to cover D is bounded by $N_{r_c}(D)$, the metric entropy, which is relatively small compared to $|D|$, the number of entries in the database (Figure 3). In contrast, if the points were uniformly distributed about the Cartesian space, $N_{r_c}(D)$ would be larger.

A second key insight is the biological datasets have low fractal dimension.[38] That is, within some range of radii r_1 and r_2 about an arbitrary point in the database D, the fractal dimension d is $d = \frac{(\log(n_2/n_1)}{(\log\ (r_2/r_1)}$, where n_1 and n_2 are the number of points within r_1 and r_2 respectively (Figure 3).

Cluster-based search, as exemplified by “compressive omics”—the use of compression to accelerate analysis— can perform approximate search within a radius r of a query q on a database D with fractal dimension d and metric entropy k at the scale r_c in time proportional to

$$O\left(\underbrace{k}_{\text{metric entropy}} + \overbrace{|B_D(q,r)|}^{\text{output size}} \underbrace{\left(\frac{r+2r_c}{r}\right)^d}_{\text{scaling factor}}\right),$$

where $B_D(q, r)$ refers to the set of points in D contained within a ball of radius r about a point q.

Given this formalization, the ratio $\frac{(|\mathrm{D}|)}{k}$ provides an estimate of the speedup factor for the coarse search component compared to a full linear search. The time complexity of the fine search is exponential in the fractal dimension d, which can be estimated globally by sampling the local fractal dimension over a dataset. The accompanying table provides the fractal dimension d sampled at typical query radii, as well as the ratio $\frac{(|\mathrm{D}|)}{k}$, for nucleotide sequence, protein sequence, protein structure, and chemical compound databases.

Biological datasets exhibit redundancy, and are constrained to subspaces by physical laws; that is, the vast majority of enumerable sequences and structures do not exist because they are not advantageous (or at least, have not been selected for by evolution). This combination results in low fractal dimension and low metric entropy relative to the size of the dataset, which suggests that “compressive omics” will provide the ability for computation to scale sublinearly with massively growing data.

The Age of Compressive Algorithms

We are entering the age of compressive algorithms, which make use of this completely different paradigm for the structure of biological data. Seeking to take advantage of the redundancy inherent in genomic sequence data, Loh, Baym and Berger[22] introduced *compressive genomics*, an approach that relies on compressing data in such a way that the desired computation (such as BLAST search) can be performed in the compressed representation. Compressive genomics is based on the concept of *compressive acceleration*, which relies on a two-stage search, referred to as *coarse* and *fine* search. Coarse search is performed only on the coarse, or representative, subsequences that represent unique data. Any representative sequence within some threshold of the query is then expanded into all similar sequences it represents; the fine search is over this (typically small) subset of the original database. This approach provides orders-of-magnitude runtime improvements to BLAST nucleotide[22] and protein[12] search; these runtime improvements increase as databases grow.

The CORA read mapper[37] applies a mid-size *l*-mer based read-compression approach with a compressive indexing of the reference genome (referred to as a homology table). CORA, like caBLAST (compressively accelerated BLAST)[22] and caBLASTP,[12] accelerates existing tools (in this case, read mappers including BWA or Bowtie2) by allowing them to operate in a compressed space, and relies on a coarse and a fine phase. In contrast, short seed-clustering schemes, such as those used in Masai[33] and MrsFAST,[3] conceptually differ from CORA in that those schemes aim to accelerate only the seed-to-reference matching step. Thus, there is a subsequent seed-extension step, which is substantially more costly and still needs to be performed for each read and mapping individually, even when seeds are clustered. Through its *l*-mer based read compression model, CORA is able to accelerate and achieve asymptotically sublinear scaling for both the seed-matching and seed-extension steps within coarse-mapping, which comprises the major bulk of the read-mapping computation. Traditionally, *k*-mers refer to short substrings of fixed length (often, but not necessarily, a power of two) used as “seeds” for longer sequence matches. CORA uses much longer *k*-mers (for example, 33–64 nucleotides long), and links each one to its neighbors within a small Hamming or Levenshtein distance. The term *l*-mer distinguishes these substrings from typically short *k*-mers.

In the area of metagenomic search, the recently released MICA[38] demonstrates the compressive-acceleration approach of caBLAST[22] and caBLASTP[12] is largely orthogonal to alphabet-reduction and indexing approaches. MICA applies the compressive-acceleration framework to the state-of-the-art DIAMOND,[7] using it for its “coarse search” phase and a user’s choice of DIAMOND or BLASTX for its “fine search” phase; MICA demonstrates nearly order-of-magnitude run-time gains over the highly optimized DIAMOND, comparable to that of caBLASTP over BLASTP.

Compressive genomics[22] has been generalized and adapted to non-sequence spaces as well, and coined “compressive omics.” One such example is chemogenomics. Applying a compressive acceleration approach, Ammolite[38] accelerates SMSD search by an average of 150× on the PubChem database. Another example is esFrag-Bag,[38] which clusters proteins based on the cosine distance or Euclidean distance of their bag-of-words vectors, further accelerating FragBag’s running time by an average of 10×.

The compressive omics approach can, in some cases, come at the cost of accuracy. However, these cases are well defined. Compressive omics never results in false positives (with respect to the naïve search technique being accelerated), because the fine search phase applies the same comparison to the candidates as the naïve approach. Furthermore, when the distance function used for comparisons is a metric—more specifically, when it obeys the triangle inequality—false negatives will also never occur. Yet, in practice, non-metric distance functions are used, such as E-values in BLAST or cosine distance in esFragBag, and thus false negatives can occur. Fortunately, these error rates are low, and recall better than 90% has been demonstrated.[12,22,38]

Conclusion

The explosion of biological data, largely due to technological advances such as next-generation sequencing, presents us with challenges as well as opportunities. The promise of unlocking the secrets of diseases such as cancer, obesity, Alzheimer's, autism spectrum disorder, and many others, as well as better understanding the basic science of biology, relies on researchers' ability to analyze the growing flood of genomic, metagenomic, structural, and interactome data.

The approach of compressive acceleration,[22] and its demonstrated ability to scale with the metric entropy of the data,[38] while providing orthogonal benefits to many other useful indexing techniques, is an important tool for coping with the deluge of data. The extension of this compressive acceleration approach to metagenomics, NGS read mapping,[37] and chemogenomics suggests its flexibility. Likewise, compressive storage for these applications can be shown to scale with the information-theoretic entropy of the dataset.[38]

The field of computational biology must continue to innovate, but also to incorporate the best ideas from other areas of computer science. For example, the compressive acceleration approach bears similarity to a metric ball tree, first described in the database community over 20 years ago;[35] however, the latter does not allow one to analyze performance guarantees in terms of metric entropy and fractal dimension. Other ideas from image processing, computational geometry,[18] sublinear-time algorithms,[30] and other areas outside of biology are likely to bear fruit. It is also likely that algorithmic ideas developed within computational biology will become useful in other fields experiencing a data deluge, such as astronomy or social networks.[34]

Biological data science is unique for two primary reasons: biology itself—even molecular biology—predates the information age, and "nothing in biology makes sense except in light of evolution."[13] Not only have biologists developed a diverse array of experimental techniques, but the data derives from astoundingly complex processes that themselves are driven by evolution. It is through the development of algorithms that leverage the structure of biological data that we can make sense of biology in light of evolution.

Acknowledgments

This work is supported by the National Institutes of Health, under grant GM108348. Y.W.Y. is also supported by a Hertz Fellowship.

References

1. 1000 Genomes Project Consortium et al. An integrated map of genetic variation from 1,092 human genomes. Nature. 2012; 491(7422):56–65. [PubMed: 23128226]
2. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of Molecular Biology. 1990; 215(3):403–410. [PubMed: 2231712]
3. Berger B, Peng J, Singh M. Computational solutions for omics data. Nature Reviews Genetics. 2013; 14(5):333–346.

4. Bonfield JK, Mahoney MV. Compression of FASTQ and SAM format sequencing data. PLoS ONE. 2013; 8(3):e59190. [PubMed: 23533605]

5. Bredel M, Jacoby E. Chemogenomics: An emerging strategy for rapid target and drug discovery. Nature Reviews Genetics. 2004; 5(4):262–275.

6. Bruijn DN. A combinatorial problem. Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A. 1946; 49(7):758.

7. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nature Methods. 2015; 12(1):59–60. [PubMed: 25402007]

8. Candes EJ, Tao T. Decoding by linear programming. IEEE Transactions on Information Theory. 2005; 51(12):4203–4215.

9. Cao M, Zhang H, Park J, Daniels NM, Crovella ME, Cowen LJ, Hescott B. Going the distance for protein function prediction: A new distance metric for protein interaction networks. PLoS ONE. 2013; 8(10)

10. Chindelevitch L, Trigg J, Regev A, Berger B. An exact arithmetic toolbox for a consistent and reproducible structural analysis of metabolic network models. Nature Communications. 2014; 5

11. Cho, H., Berger, B., Peng, J. Research in Computational Molecular Biology. Vol. 2015. Springer; Diffusion component analysis: Unraveling functional topology in biological networks; p. 62-64.

12. Daniels NM, Gallant A, Peng J, Cowen LJ, Baym M, Berger M. Compressive genomics for protein databases. Bioinformatics. 2013; 29:i283–i290. [PubMed: 23812995]

13. Dobzhansky T. Nothing in biology makes sense except in the light of evolution. 1973

14. Forsberg KJ, Reyes A, Wang B, Selleck EM, Sommer MO, Dantas G. The shared antibiotic resistome of soil bacteria and human pathogens. Science. 2012; 337(6098):1107–1111. [PubMed: 22936781]

15. Hach F, Hormozdiari F, Alkan C, Hormozdiari F, Birol I, Eichler EE, Sahinalp SC. mrsFAST: A cache-oblivious algorithm for short-read mapping. Nature Methods. 2010; 7(8):576–577. [PubMed: 20676076]

16. Hach F, Sarra I, Hormozdiari F, Alkan C, Eichler EE, Sahinalp SC. mrsFAST-Ultra: a compact, SNP-aware mapper for high-performance sequencing applications. Nucleic Acids Research. 2014 gku370.

17. Hart Y, Sheftel H, Hausser J, Szekely P, Ben-Moshe NB, Korem Y, Tendler A, Mayo AE, Alon U. Inferring biological tasks using Pareto analysis of high-dimensional data. Nature Methods. 2015; 12(3):233–235. [PubMed: 25622107]

18. Indyk, P., Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. Proceedings of the 13th Annual ACM Symposium on Theory of Computing; ACM; 1998. p. 604-613.

19. Janda JM, Abbott SL. 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: pluses, perils, and pitfalls. J. Clinical Microbiology. 2007; 45(9):2761–2764.

20. Jardine N, van Rijsbergen CJ. The use of hierarchic clustering in information retrieval. Information Storage and Retrieval. 1971; 7(5):217–240.

21. Liao C-S, Lu K, Baym M, Singh R, Berger B. IsoRankN: Spectral methods for global alignment of multiple protein networks. Bioinformatics. 2009; 12:i253–i258.

22. Loh P-R, Baym M, Berger B. Compressive genomics. Nature Biotechnology. 2012; 30(7):627–630.

23. MacFabe DF. Short-chain fatty acid fermentation products of the gut microbiome: Implications in autism spectrum disorders. Microbial Ecology in Health and Disease. 2012; 23

24. Marco-Sola S, Sammeth M, Guigó R, Ribeca P. The gem mapper: Fast, accurate and versatile alignment by filtration. Nature Methods. 2012; 9(12):1185–1188. [PubMed: 23103880]

25. Marx V. Biology: The big challenges of big data. Nature. 2013; 498(7453):255–260. [PubMed: 23765498]

26. Ochoa I, Asnani H, Bharadia D, Chowdhury M, Weissman T, Yona G. QualComp: A new lossy compressor for quality scores based on rate distortion theory. BMC bioinformatics. 2013; 14(1): 187. [PubMed: 23758828]

27. Patro R, Kingsford C. Data-dependent bucketing improves reference-free compression of sequencing reads. Bioinformatics. 2015

28. Prat Y, Fromer M, Linial N, Linial M. Recovering key biological constituents through sparse representation of gene expression. Bioinformatics. 2011; 5:655–661.

29. Rahman SA, Bashton M, Holliday GL, Schrader R, Thornton JM. Small molecule subgraph detector (SMSD) toolkit. J. Cheminformatics. 2009; 1(1):1–13.

30. Rubinfeld R, Shapira A. Sublinear time algorithms. SIAM J. Discrete Mathematics. 2011; 25(4): 1562–1588.

31. Schatz MC, Langmead B, Salzberg SL. Cloud computing and the DNA data race. Nature Biotechnology. 2010; 28(7):691–693.

32. Singh R, Xu J, Berger B. Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences. 2008; 105(35):12763–12768.

33. Siragusa E, Weese D, Reinert K. Fast and accurate read mapping with approximate seeds and multiple backtracking. Nucleic Acids Research. 2013; 41(7):e78. [PubMed: 23358824]

34. Stephens ZD, et al. Big data: Astronomical or genomical? PLoS Biol. 2015; 13(7):e1002195. [PubMed: 26151137]

35. Uhlmann JK. Satisfying general proximity/similarity queries with metric trees. Information Processing Letters. 1991; 40(4):175–179.

36. Weinstein JN, et al. The cancer genome atlas pan-cancer analysis project. Nature Genetics. 2013; 45(10):1113–1120. [PubMed: 24071849]

37. Yorukoglu D, Yu YW, Peng J, Berger B. Compressive mapping for next-generation sequencing. Nature Biotechnology. 2016; 4:374–376.

38. Yu YW, Daniels N, Danko DC, Berger B. Entropy-scaling search of massive biological data. Cell Systems. 2015; 1(2):130–140. [PubMed: 26436140]

39. Yu YW, Yorukoglu D, Peng J, Berger B. Quality score compression improves genotyping accuracy. Nature Biotechnology. 2015; 33(3):240–243.

40. Zhao Y, Tang H, Ye Y. RAPSearch2: A fast and memory-efficient protein similarity search tool for next-generation sequencing data. Bioinformatics. 2012; 28(1):125–126. [PubMed: 22039206]

key insights

- There is a lot of commonality in sequences and other biological data— even more redundancy than in a text file of the English language.
- This means we can take advantage of compression algorithms that exploit that commonality and represent many sequences by only a few bits.
- Of course, we are dealing with a massive amount of data so that compression becomes important for efficiency.
- We highlight recent research that capitalizes on structural properties of biological data—low metric entropy and fractal dimension—to allow us to design algorithms that run in sublinear time and space.

Definitions

Chemogenomics: Computational study of drugs and their targets based on chemical structure and function.

Metagenomics: Study of the many genomes that make up a particular environment.

Shotgun sequencing: Modern genomic sequencing, which chops DNA into many short pieces

Homology search: Determining the function, structure, or identity of a gene sequence by locating similar sequences within an annotated database.

Transcriptome: Transcribed RNA from a genome, which results in protein production.

BLAST: Standard biological sequence similarity search tool.

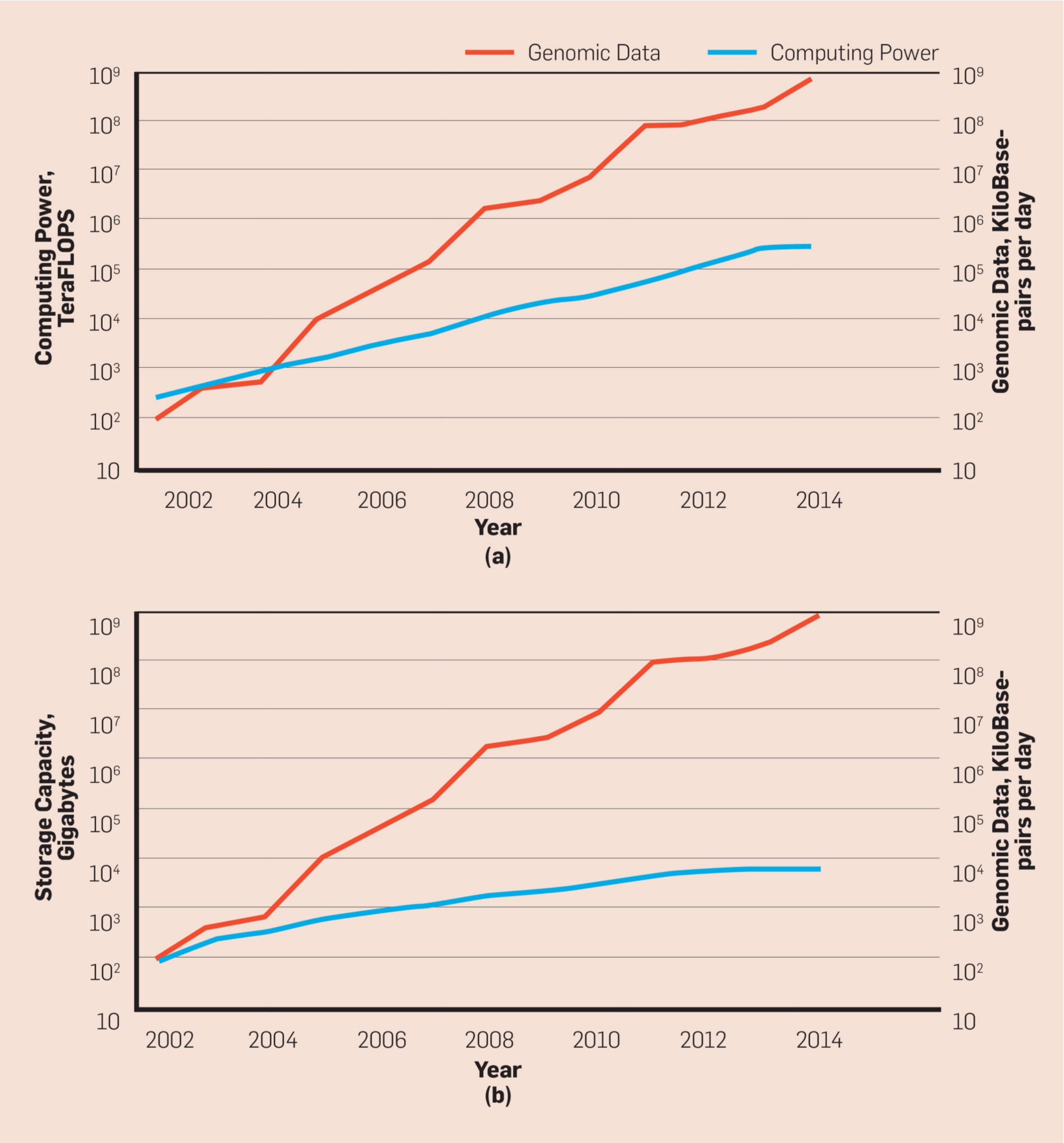
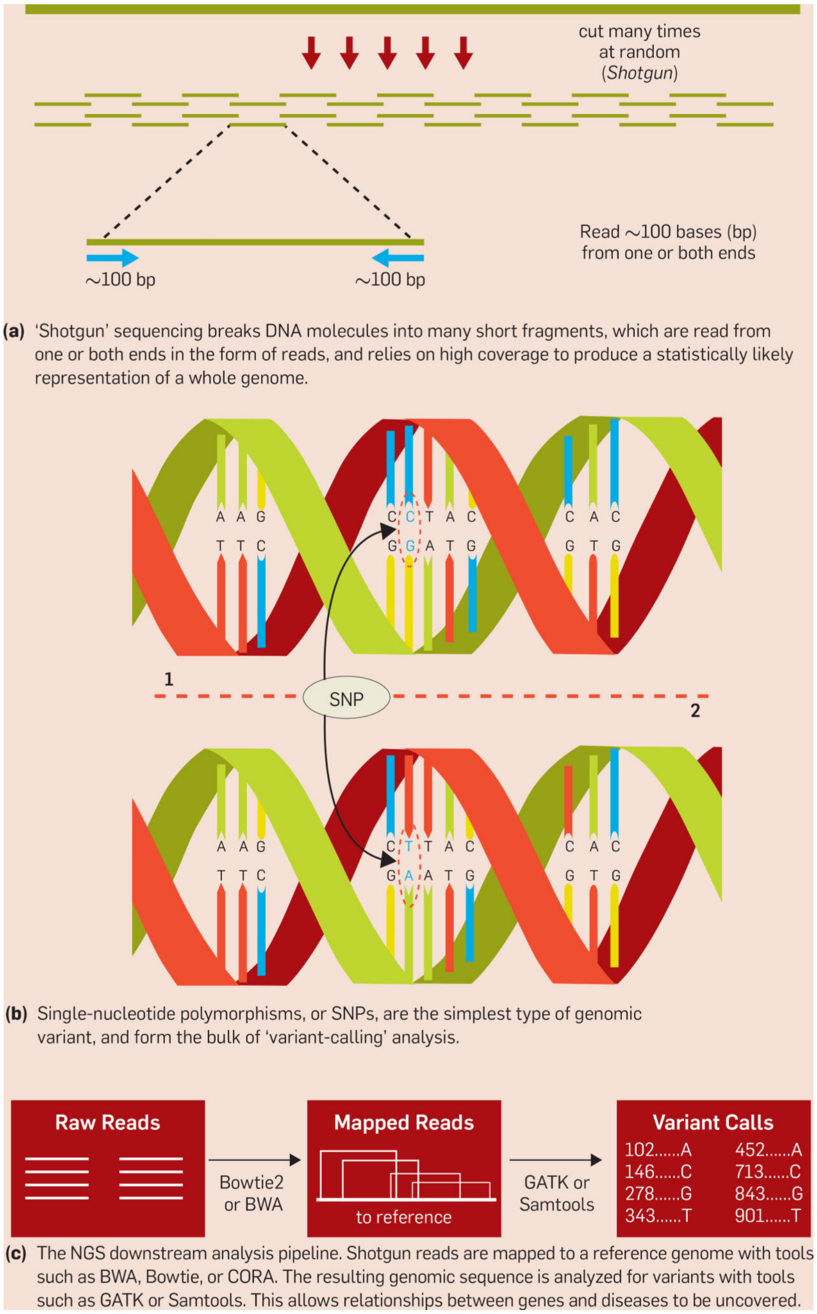


Figure 1.

(a) Moore's and (b) Kryder's laws contrasted with genomic sequence data.



(a) 'Shotgun' sequencing breaks DNA molecules into many short fragments, which are read from one or both ends in the form of reads, and relies on high coverage to produce a statistically likely representation of a whole genome.

(b) Single-nucleotide polymorphisms, or SNPs, are the simplest type of genomic variant, and form the bulk of 'variant-calling' analysis.

(c) The NGS downstream analysis pipeline. Shotgun reads are mapped to a reference genome with tools such as BWA, Bowtie, or CORA. The resulting genomic sequence is analyzed for variants with tools such as GATK or Samtools. This allows relationships between genes and diseases to be uncovered.

Figure 2.

The next-generation sequencing (NGS) pipeline.

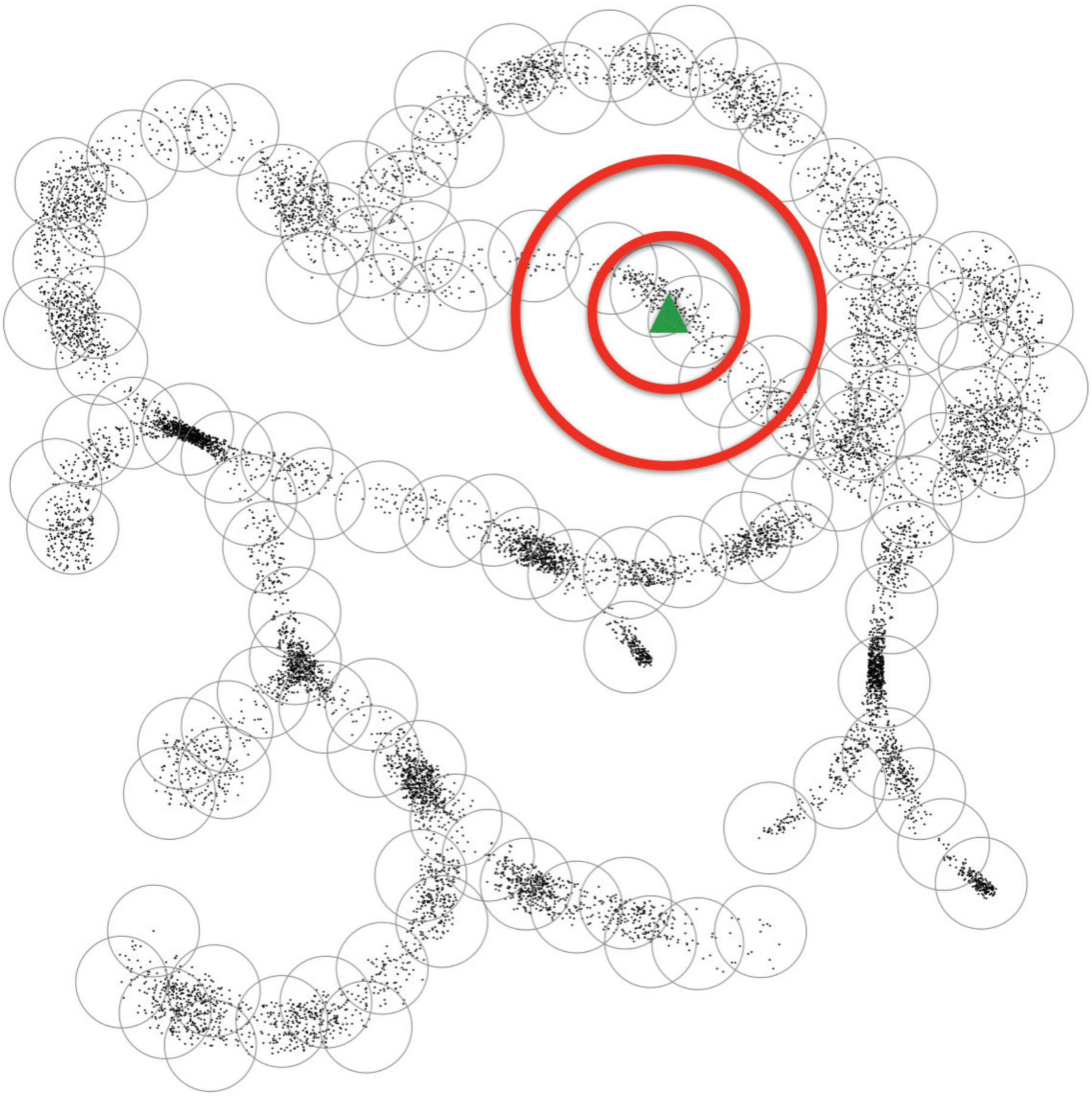


Figure 3.

Cartoon depiction of points in an arbitrary high-dimensional space, as might arise from genomes generated by mutation and selection during the course of evolution.

Although high dimensional locally, at the global scale of covering spheres, the data cloud looks nearly 1-dimensional, which enables entropy scaling of similarity search. Clusters cover the data points but do not cover unoccupied regions of space. The green triangle represents a query, with two concentric search radii (red circles) around it. Thanks to low fractal dimension, the large circle does not contain vastly more points than the small circle.

Metric-entropy ratio (ratio of clusters to entries in database) and fractal dimension at typical search radii for four datasets

Metric-entropy ratio gives an estimate of the acceleration of coarse search with respect to naïve search, and as long as fractal dimension is low, coarse search should dominate total search time. NCBI's non-redundant 'NR' protein and 'NT' nucleotide sequence databases are from June 2015. Protein Data Bank (PDB) is from July 2015. PubChem is from October 2013.

Dataset	Metric-entropy ratio	Fractal dimension
Nucleotide sequences (NCBI NT)	7:1	1.5
Protein sequences (NCBI NR)	5:1	1.6
Protein structure (PDB)	10:1	2.5
Chemical structure (PubChem)	11:1	0.2