THE GENOME QUESTION: MOORE VS. JEVONS

B. MISHRA[1]

JANUARY 11, 2012

[1] It is often said that genomics science is on a Moore's law, growing exponentially in data throughput, number of assembled genomes, lowered cost, etc.; and yet, it has not delivered the biomedical promises made a decade ago: personalized medicine; genomic characterization of diseases like cancer, schizophrenia, and autism; bio-markers for common complex diseases; prenatal genomic assays, etc. What share of blame for this failure ought to be allocated to computer science (or computational biology, bioinformatics, statistical genetics, etc.)? How can the Indian computational biology community lead genomics science to rescue it from the current impasse? What are the computational solutions to these problems? What should be our vision of computational biology in the coming decade?

[1] (Professor of Mathematics and Computer Science, Courant Institute, NY, USA)

[2]

[2] Dedicated to a mentor and friend, Bill Wulf, who taught me languages, architectures and compilers, but much more importantly, how to "stop worrying and love Moore's Law!"

Jevons' Paradox

William Stanley Jevons (1835 – 1882), a British computer scientist, statistician, logician and economist, is usually remembered, along with Menger and Walras, for his work in economics on marginal utility theory of value – and only occasionally for his work on an early logical computer, the "Logic Piano," constructed in 1869. He is also immortalized by a paradox, which bears his name, first identified in his 1865 book "The Coal Question" [3]. Jevons observed that, after James Watt introduced his more efficient coal-fired steam engine, somewhat counter-intuitively, England's consumption of coal skyrocketed – notwithstanding the savings from higher efficiency. Because of innovations due to Watt and others, which improved the engines' efficiency over Thomas Newcomen's existing design, rapidly coal became much more cost-effective as a source of energy and the steam engine found many more hitherto unanticipated applications. Consequently, the total coal consumption increased quickly – but not necessarily always with a beneficial effect. This paradoxical rebound effect [4] came to be called a "backfire."

[3] W.S. Jevons. *The Coal Question*. Macmillan and Co., 1865.

[4] B. Alcott. Jevons' paradox. *Ecol. Econ.*, 54: 9–21, 2005.

Over the last few years, a genomic version of "Jevons' paradox" seems to be playing out in the arena of biotechnology. With the advent of next-generation sequencing technologies, and with rapidly decreasing cost of optical or electronic detection systems

that they require, it has become exponentially easier, faster and cheaper to acquire – in a matter of weeks, if not days – hundreds of times coverage in sequence reads of thousands of human-sized genomes. The massive amount of data generated by these machines has kept lab technicians, bioinformaticists, systems managers, computer architects and computer scientists increasingly busy, diverting their attention from fundamental statistical, algorithmic and ultimately, biomedical innovations [5]. As they spend more and more time shifting, shuffling, storing and stockpiling massive amounts of sequence data spewing out of marginal-quality second- and third-generation sequencing platforms, they find themselves embroiled in a Jevonian backfire – the torrential amount of data appears to have paradoxically dried up the biological insights and biomedical outcomes. The genomic revolution appears to have been postponed – at least, for the time being. A shadow of doubt has been cast on the progress so far; one wonders: whether the human genome project has produced a reference that correctly reflects the complexity of genome rearrangements; whether Hapmap captures the most meaningful genomic variants; or whether the disease markers obtained from large-scale GWAS (genome-wide association studies) correctly interpret the etiology of common human diseases. Uncannily, biotechnology's Moore's law is now stuck in a Jevons paradox. It has been said, most likely in jest, that biotechnology has been on an "Inverse-Moore's Law" – every eighteen months or so, despite (or because of) exponential growth in genomics data, the biologists are becoming twice less insightful.

[5] See "DNA Sequencing Caught in Deluge of Data," by A. Pollack, Business Day, New York Times, November 30 2011, & "Storage Saga," by M. Dublin, Genome Technology, Genome Web, December 2011/January 2012.

Following is a short list of critical issues that the field faces now:

***Data Storage*:** With the advent of next-generation sequencing technologies around 2008, the throughput from the sequencing platforms have outpaced both computer and storage technologies. Hard-disk technologies develop under a principle called Kryder's law, which states that storage disk density doubles annually and that the cost of storing one byte of data is halved every fourteen months as a result. On the other hand, the per-base cost of sequencing is dropping by half about every five months and is likely to continue at that rate for the near future. However, since individual and populations of genomes are highly structured, one could exploit this structure in a Bayesian manner to bound the amount of sampling needed (number of individuals as well as the coverage needed for each individual genome) and the degree to which the genomic data can be compressed (loss-lessly or in lossy compression; see TotalRecaller [6], which combines base-calling, alignment and variant-calling in one step). Ideas from probabilistic analysis (e.g., 0-1-laws) [7], rate-distortion

[6] F. Menges, G. Narzisi, and B. Mishra. Totalrecaller: Improved accuracy and performance via integrated alignment & base-calling. *Bioinformatics*, 5(7), 2011.

[7] T.S. Anantharaman and B. Mishra. Genomics via optical mapping (i): 0-1 laws for single molecules. *Technical Note: Unpublished Manuscript*, 2000. URL `http://www.cs.nyu.edu/mishra/PUBLICATIONS/04.gvom.pdf`.

theory [8], algorithms based on Block-Sorting compression and related opportunistic data structures [9] can provide the needed solution.

***Error-Correction*:** Despite many technological advances, all next-generation sequencing platforms tend to be significantly more error-prone. The errors have many sources, but are primarily caused by loss of synchronization in the bio-chemical cycles, which operate in synchrony on a small number of clonal copies of the DNA being sequenced [10], [11]. As the feature sizes (wells, beads, and bound DNAs) are reduced in order to increase the throughput, these errors become further exacerbated, and traditional base callers fail after reading only a few hundred base pairs. Other error sources leading to homo-polymer compression and chimerisms impose even more difficulties. Novel Bayesian base callers such as TotalReCaller [12], mentioned earlier, have addressed many of these problems quite well, but still leave room for many additional improvements. The current trend of using "error correctors" or "gap fillers," as a pre-processing step in sequence assembly, consumes a disproportionately large fraction of assembly-compute-cycles, while producing improvement of dubious quality. Since the assembler and its error corrector modules run in the cloud, this strategy also entails a large amount of unnecessary data movement.

***Unused and Low Effective-Coverage*:** Because of the size limitations and increased error rates in the sequence reads from next-generation sequencers, the assemblers require a higher overlap threshold ratio θ, thus reducing the effective coverage $c_e = c(1-\theta)$, where $c = NL/G =$ coverage, $N =$ number of reads, $L =$ read length and $G =$ the genome size [13]. Thus while the improvement in the total achievable coverage under biotech's Moore's law is breathtakingly impressive, the improvement in effective coverage is only modest. Furthermore, most assemblers, developed after 2008 to specifically handle next-generation sequencing data [14], [15], [16], tame their computational complexity by only computing overlaps with exact matches (which can be computed quickly with prefix trees) and structuring shorter k-mers from the reads in a deBruijn graph, thus worsening the data/coverage loss. Specific heuristics in these algorithms, developed to deal with "dead ends" and "p-bubbles" are symptomatic of the massive amount of data leakage that must be tolerated by these Eulerian-path algorithms.

***Missingness*:** Because of historic reasons, the algorithmic strategies still favored by most are based on shot-gun assembly

[8] T. Berger. *Rate Distortion Theory*. Prentice-Hall, 1971.

[9] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *FOCS*, pages 390–398, 2000.

[10] Y. Erlich, P.P. Mitra, M. delaBastide, W.R. McCombie, and G.J. Hannon. Alta-Cyclic: a self-optimizing base caller for next-generation sequencing. *Nat. Methods*, 5:679–682, Aug 2008.

[11] Illumina. De novo assembly using illumina reads. *Technical Note: sequencing*, 2010. URL `http://www.illumina.com`.

[12] F. Menges, G. Narzisi, and B. Mishra. Totalrecaller: Improved accuracy and performance via integrated alignment & base-calling. *Bioinformatics*, 5(7), 2011.

[13] E.S. Lander and M.S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2(231), 1988.

[14] R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, S. Li, H. Yang, J. Wang, and J. Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20(2):265–272, 2010.

[15] J.T. Simpson, K. Wong, S.D. Jackman, J.E. Schein, S.J. Jones, and A. Birol. Abyss: A parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009.

[16] D.R. Zerbino and E. Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18 (5):821–829, 2008.

– with few post-processing modules added, like an afterthought, at the end for scaffolding and validation using mate pairs. Consequently, the assembly produced is usually genotypic, lacking any reliable characterization of rearrangements or haplotypic phasing. Very few assemblers have built-in mechanisms to utilize more informative long-range information (e.g., from optical maps, dilution mapping or strobed sequences) in order to perform haplotypic and self-validating assembly – the only interesting exception being SUTTA [17], [18], [19]. The missingness has led to unsatisfactory results, when such genome assemblies are used in GWAS (genome-wide association studies), since they adversely affect population stratification, null models (distorted by Yule-Simpson effects), resulting errors in p-values and subsequent multiple-hypothesis-testing corrections.

***Correctness*:** Finally, as these new technologies have spurred more and more assembled genome references for thousands of organisms, their assembly accuracy remains uncertain. Even the human reference genome, after some thirty-seven builds, is still only genotypic and is suspected to have many unrecognized rearrangement errors. The usual metrics, such as N50, routinely used to characterize the strength and accuracy of assemblers, has been found to be rather misleading [20]. Recent studies have shown that the genome simulators used in characterizing the genome assemblers are unreliable. Although technologies to produce long-range, albeit lower-resolution, genomic information have been available for more than a decade, none of them have been used in any meaningful way to validate genome references in all but a handful of microbial genomes [21], [22], [23], [24], [25] and [26].

In summary, one may pause here and question whether genomics is really better off with the Moore's law that it has spawned. In the rest of the paper, we attempt to assume a tone of measured optimism.

A Genome Operating System

Following our optimism, below, we suggest several innovations, all relying upon intelligent Bayesian priors in order to perform data compression, error correction, haplotypic phasing and assembly validation *as early and as eagerly as possible* in the sequencing pipeline. Thus instead of moving the raw data to a central repository (e.g., a cloud), where model-driven algorithms perform data interpretation, correction and assimilation, the models should

[17] G. Narzisi and B. Mishra. Scoring-and-unfolding trimmed tree assembler: concepts, constructs and comparisons. *Bioinformatics*, 27 (2):153–160, 2011b.

[18] G. Narzisi. *Scoring-and-Unfolding Trimmed Tree Assembler: Algorithms for Assembling Genome Sequences Accurately and Efficiently*. PhD thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, 2011.

[19] G. Narzisi and B. Mishra. Comparing de novo genome assembly: The long and short of it. *PLoS ONE*, 6(4):e19175, 2011a.

[20] F. Vezzi, G. Narzisi, and B. Mishra. Feature-by-feature – evaluating de novo sequence assembly. *PLoS ONE*, page (In Press), 2012.

[21] B. Mishra. Optical mapping. *Encyclopedia of the Human Genome*, 4:448–453, 2003.

[22] C. Aston, D.C. Schwartz, and B. Mishra. Optical mapping and its potential for large-scale sequencing projects. *Trends in Biotechnology*, 17:297–302, 1999.

[23] J. Lin et al. Whole genome shotgun optical mapping of deinococcus radiodurans. *Science*, 285(5433):1558–1562, 1999a.

[24] Z. Lai et al. A shotgun sequence-ready optical map of the whole plasmodium falciparum genome. *Nature Genetics*, 23(3): 309–313, 1999b.

[25] A. Lim et al. Shotgun optical maps of the whole escherichia coli o157:h7 genome. *Genome Research*, 11(9):1584–1593, 2001.

[26] T.S. Anantharaman, V. Mysore, and B. Mishra. Fast and cheap genome wide haplotype construction via optical mapping. *The Pacific Symposium on Biocomputing*, PSB 2005:385–396, 2005.

be moved to the periphery – getting as close to the sequencing machines as possible.

Thus, in this architecture, a suitably modified sequencing machine may hold a compressed version of the draft or finished reference genome (or its approximation that is statistically indistinguishable from the true reference) to perform base calling with both error correction and data compression (using a Δ-modulation scheme); since pattern recognition and variant calling will come for free, its usefulness as a clinical machine should be obvious. Similarly, the error processes corrupting the sequence reads, informative structures occurring in the genome, and the long-range information from the genome can all be encoded in certain likelihood-based score functions, which could then be made available to the sequence assembler so it can build a self-validating assembly of the genome. Such an assembler can be easily formulated as a constrained global optimization problem, as has been done in the pipeline of NYU's Bioinformatics Group consisting of TotalReCaller, SUTTA, "Long-Range-Map-Scores" and "ICA-Feature-Scores;" the current implementation has been validated with mate pairs, and similar scores based on dilution maps, optical maps or clone libraries are also applicable.

Within our framework, where the optimization algorithm's architecture is separated from the domain-, genome- and sequencing-device-specific properties (via well-selected priors and ensuing score and penalty functions), the pipeline is capable of handling multiple technologies, being technologically agnostic as well as rapidly evolvable. Below, we delve into the details of three of the pipeline's main modules.

- TotalReCaller [27] aims to improve base calling quality by interpreting the analog signals from sequencing machines, while simultaneously aligning the sequence reads to a source reference (draft or finished) genome, whenever available, to reduce the error rate.

- SUTTA [28], [29] is a self-validating sequence assembler, based on a flexible branch-and-bound [30] framework that forcefully and quickly eliminates incorrect solutions (i.e., implausible layouts). To achieve this goal, SUTTA relies on technology-agnostic score functions that enable combining data from multiple sources and distinct technologies.

- Feature-Response Curves (FRC) [31] have been designed to evaluate the relative accuracies, coverages and contig sizes of the outputs from different assembly pipelines. PCA and ICA (principal and independent component analysis, respectively) have been used to characterize and select the most

[27] F. Menges, G. Narzisi, and B. Mishra. Totalrecaller: Improved accuracy and performance via integrated alignment & base-calling. *Bioinformatics*, 5(7), 2011.

[28] G. Narzisi and B. Mishra. Scoring-and-unfolding trimmed tree assembler: concepts, constructs and comparisons. *Bioinformatics*, 27 (2):153–160, 2011b.

[29] G. Narzisi. *Scoring-and-Unfolding Trimmed Tree Assembler: Algorithms for Assembling Genome Sequences Accurately and Efficiently*. PhD thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, 2011.

[30] A.H. Land and A.G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[31] G. Narzisi and B. Mishra. Comparing de novo genome assembly: The long and short of it. *PLoS ONE*, 6(4):e19175, 2011a.

informative features that can evaluate the correctness of the assembly [32].

[32] F. Vezzi, G. Narzisi, and B. Mishra. Feature-by-feature – evaluating de novo sequence assembly. *PLoS ONE*, page (In Press), 2012.

TotalReCaller

TotalReCaller combines the knowledge from sequencers' raw intensity data with information from a reference genome (when available). In other words, it generates the most plausible m-base string (out of 4^m possibilities) that is most likely to have generated the channel intensity (analog) data, and also most likely to have originated at some location of the reference genome (and spanning m bases). Like many global combinatorial optimization problems, TotalReCaller tames the worst-case exponential complexity of the implementation by using a beam search [33] strategy (an adaptation of the branch-and-bound [34] method).

[33] R. Bisiani. *Beam search*, pages 56–58. Wiley & Sons, 1987.

[34] A.H. Land and A.G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

For this purpose TotalReCaller relies on a base-by-base alignment algorithm, based on the Ferragina-Manzini search, which serves as a feedback for a linear error model, resulting in this novel approach to base calling [35].

[35] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *FOCS*, pages 390–398, 2000.

Differently from previously published base callers, TotalReCaller uses a completely new strategy to recover each base of the sequence from the raw sequencing data. Specifically, this strategy is used to concurrently extend multiple high-quality reads that are immediately validated not only by the intensity signals but also by the likely alignments to a reference genome (thus the genome provides a weak prior to a Bayesian inference). This scheme builds on a rigorously defined Bayesian score function that accounts for both — thereby resulting in a single score to quantify the quality of a given sequence read. Since, by Bayes' theorem, the conditional probability of a base B, given an intensity $\mathbf{X}_k$ (in the kth cycle) is

$$P_k(B \mid \mathbf{X}_k) = \frac{P_k(\mathbf{X}_k \mid B)P_k(B)}{P_k(\mathbf{X}_k)} \quad \text{with} \quad B \in \{A, C, G, T\} \tag{1}$$

$$= \frac{P_k(\mathbf{X}_k \mid B)P_k(B)}{P_k(\mathbf{X}_k \mid B)P_k(B) + P_k(\mathbf{X}_k \mid \neg B)P_k(\neg B)} \tag{2}$$

$$= \frac{1}{1 + \frac{P_k(\mathbf{X}_k|\neg B)}{P_k(\mathbf{X}_k|B)} \cdot \frac{P_k(\neg B)}{P_k(B)}}, \tag{3}$$

it leads to a simplified score function:

$$f_{score} = \underbrace{\log\left(\frac{P_k(\mathbf{X}_k \mid B)}{P_k(\mathbf{X}_k \mid \neg B)}\right)}_{\text{intensity-based score}} + w_{align} \cdot \underbrace{\log\left(\frac{P_k(B)}{P_k(\neg B)}\right)}_{\text{alignment-based score}} \tag{4}$$

In order to execute the base calling task, TotalReCaller implements four different components that are described in detail

elsewhere [36]: (1) linear error model; (2) base-by-base sequence alignment; (3) beam search read extension; and, finally, (4) score function. Note that TotalReCaller can be implemented directly in hardware (GPU or FPGA), residing close to (or embedded in) the sequencing platform. Furthermore, since it computes the alignment with respect to a reference directly, it only needs to output a compressed data stream, containing the location of alignments and base pair differences (including SNV's and indels) for the receiver to be able to reconstruct the exact sequence. In the most advanced implementations, TotalReCaller may only output those SNPs that are clinically relevant, or only those de novo mutations that are still uncharacterized.

[36] F. Menges, G. Narzisi, and B. Mishra. Totalrecaller: Improved accuracy and performance via integrated alignment & base-calling. *Bioinformatics*, 5(7), 2011.

SUTTA

SUTTA, unlike the traditional heuristics-based assembly algorithms (e.g., greedy, sequencing by hybridization or overlap layout consensus), assembles each contig independently and dynamically one after another using the Branch-and-Bound (B&B) strategy. Originally developed for linear programming problems [37], B&B algorithms are well-known searching techniques applied to intractable ($\mathcal{NP}$-hard) combinatorial optimization problems. While SUTTA follows the basic idea of searching the complete space of assembly solutions, it avoids the usual caveat that explicit enumeration is practically impossible (i.e. due to exponential time and space complexity). The tactics honed by B&B are to limit the search to a smaller subspace that contains the optimum. This subspace is determined dynamically through the use of certain *well-chosen* score functions.

[37] A.H. Land and A.G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

At a high level, SUTTA's framework views the assembly problem simply as that of constrained optimization (based on a formulation of overlaps in consistent layouts): it relies on a rather simple and easily verifiable definition of feasible solutions as "consistent layouts." It generates potentially all possible consistent layouts, organizing them as paths in a "double tree" structure rooted at a randomly selected "seed" read. A path is progressively evaluated in terms of an optimality criterion, encoded by a set of score functions based on the set of overlaps along the layout.

This strategy enables the algorithm to concurrently assemble and check the validity of the layouts (with respect to various long-range information) through well-chosen constraint-related penalty functions. Complexity and scalability problems are addressed by pruning most of the implausible layouts via a *branch-and-bound* scheme. Ambiguities resulting from repeats or haplotypic dissimilarities may occasionally delay immediate pruning and force the algorithm to perform *lookahead*, but in practice, the computa-

tional cost of these events has been low. Because of the generality and flexibility of the scheme (it only depends on the underlying sequencing technologies through the choice of score and penalty functions), SUTTA is extensible, at least in principle, to deal with possible future technologies. It also allows concurrent assembly and validation of multiple layouts, thus providing a flexible framework that combines short- and long-range information from different technologies (e.g., optical or dilution mapping). In a similar manner, SUTTA can also use supervised learning to cull the most informative metrics, out of those currently popular (e.g., N50 or overlap structures, or standard amosville features [38]) to optimize a score that ensures that only the most "reasonable" layout is generated and used. Because of its reliance on a universal framework, it allows no room for (nor does it need to) *ad hoc* heuristics for error correction, gap filling, repeat masking, etc.

[38] A.M. Phillippy, M.C. Schatz, and M. Pop. Genome assembly forensics: Finding the elusive misassembly. *Genome biology*, 9:R55, 2008.

The high level SUTTA pseudocode is shown in Algorithm 1. Here, two important data structures are maintained: a forest of double trees (D-trees) $\mathcal{B}$ and a set of contigs $\mathcal{C}$. At each step a new D-tree is initiated from one of the remaining reads in $\mathcal{F}$. Once the construction of the D-tree is completed, the associated contig is created and stored in the set of contigs $\mathcal{C}$. Next the layout for this contig is computed and all its reads are removed from the set of all available reads $\mathcal{F}$. This process continues as long as there are reads left in the set $\mathcal{F}$.

Input: Set of N reads
Output: Set of contigs

```
1  B := ∅;                                              /* Forest of D-trees */
2  C := ∅;                                              /* Set of contigs */
3  F := ⋃_i^N {r_i};              /* All the available reads/fragments */
4  while (F ≠ ∅) do
5      r := F.getNextRead();
6      if ( ¬isUsed(r) ∧ ¬isContained(r) ) then
7          DT := create_double_tree(r);
8          B := B ∪ {DT};
9          Contig CTG := create_contig(DT);
10         C := C ∪ {CTG};
11         CTG.layout();                          /* Compute contig layout */
           F := F \ {CTG.reads};                  /* Remove used reads */
12     end
13 end
14 return C;
```

Figure 1: Algorithm 1: SUTTA Pseudo Code.

Finally, note that the proposed Algorithm 1 is input-order dependent. SUTTA adopts a simple ordering policy, which always selects the next unassembled read with the highest occurrence as seed for the D-tree. This strategy minimizes the extension of reads containing sequencing errors. However, empirical observations indicate that changing the order of the reads rarely affects the structure of the solutions, as the relatively longer contigs are not affected.

Scaling SUTTA to handle larger genomes, at the macro level, primarily requires a substantial speed improvement, which is achieved in following two ways: (1) improving the single-thread execution time and (2) parallelizing the basic SUTTA algorithm to the best possible extent. One of the most expensive tasks that SUTTA needs to perform is the Overlapper, which has been scaled using a divide-and-conquer approach, in which the entire data set of reads is divided into multiple "prefix trees" that can be accessed independently and in parallel. SUTTA's kernel is in the process of being redesigned and reengineered so that it can take advantage of modern multi-core microprocessor (16 to 32 core) architectures. The main innovation in the redesign involves a clever optimization of the code used in the D-tree generation, which is complicated by its need to efficiently perform *transitivity collapse*

(to keep the search tree skinny) and *lookahead* (to disambiguate repeat boundaries and haplotypic differences).

FRC

Complex genomic structures, intricate error profiles and error-prone long-range information conspire in convoluted ways to make *de novo* sequencing tasks challenging, and are usually handled by different sequence assemblers in idiosyncratic manners. Thus, it is unclear how to quantify the accuracy and contiguity of the output of a sequence assembler. This problem is further complicated by the fact that, more often than not, no (accurate) reference genome is available to assess the correctness of the assembled contigs. Furthermore, widely used metrics (such as N50 contig size), for this purpose, can be highly misleading, since they only emphasize size, poorly capturing the contig quality.

The Feature-Response Curve (FRCurve) is a novel assembly metric [39] to overcome many of these limitations. It is publicly available as an AMOS module. By analyzing the arrangement of the reads in a contig and producing a simple curve, FRC is able to evaluate and compare different assemblies and assemblers. Specifically, inspired by the receiver operating characteristic (ROC) curve, the FRCurve captures the trade-offs between contiguity (genome coverage) and quality (number of features/errors) of the assembled contigs. Features are computed using the automated assembly validation pipeline, `amosValidate`, which analyzes the output of an assembler using a suite of manually selected assembly metrics. Using `amosvalidate`, each contig is assigned a number of features that correspond to suspicious regions of the sequence. For example, in the case of mate pair checking, the `amosvalidate` tool flags regions where multiple mate pairs are mis-oriented or the insert coverage is low. Given such a set of features, the FRCurve analyzes the response (quality) of the assembler output as a function of the maximum number of possible errors (features) allowed in the contigs. Specifically, for any fixed feature threshold Φ, the contigs are sorted by size and, starting from the longest, only those contigs are tallied whose sum of features is $\leq \Phi$. For this set of contigs, the corresponding approximate genome coverage is computed, leading to a single point of the FRCurve. Since no reference sequence is used in this process, the FRCurve is particularly useful in *de novo* sequencing projects. Furthermore, separate FRCurves can be generated for each feature type, allowing the analysis of the relative strengths and weaknesses of different assemblers. A rapidly growing FRCurve usually signifies better assemblies.

Thus, FRC transparently captures the trade-offs between con-

[39] G. Narzisi and B. Mishra. Comparing de novo genome assembly: The long and short of it. *PLoS ONE*, 6(4):e19175, 2011a.

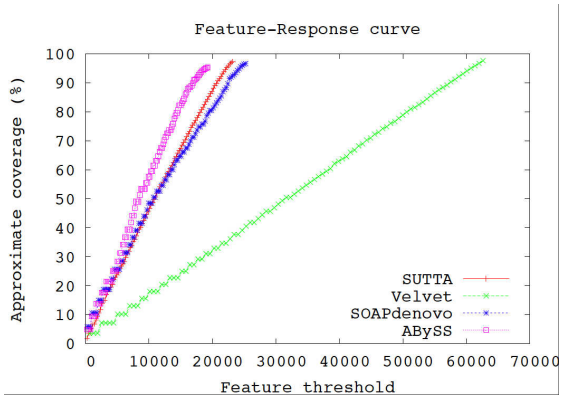

Figure 2: Feature-Response Curve (FRCurve).

tigs' quality against their size. Nevertheless, the relationship among the different features and their importance is still largely unknown, and can only be inferred empirically. In particular, FRC cannot account for correlation among the different features. In a further extension of FRC, we have recently analyzed the relationship among different features in order to better describe their relationships and their importance in gauging assembly quality and correctness. In particular, using multivariate techniques like principal and independent component analyses we were able to estimate the "excess dimensionality" of the feature space. Furthermore, principal component analysis pointed out how poorly the acclaimed N50 metric describes the assembly quality. Applying independent component analysis, it was possible to identify a subset of features that better describe the assemblers' performances. Thus, by focusing on a reduced set of highly informative features, the FRCurve can reliably describe and compare the performances of different assemblers.

Improvements in Assembly

Unlike other sequence assemblers, SUTTA does not include any error correction preprocessing step. So we designed the following pipeline to take advantage of both SUTTA and TotalReCaller capabilities [40,41]:

1. Draft Assembly: Using SUTTA (or any other sequence assembler) generate a draft assembly using the available reads.
2. Base calling & Error correction: Given the reads' intensity files and the draft assembly (generated in step 1), run TotalReCaller to generate a new set of reads with higher accuracy.
3. Sequence Assembly: Run SUTTA on the new set of reads generated in step 2 to create an improved assembly.

Steps 2 and 3 may be repeated several times in order to further improve the assembly quality, although the results presented here only use a single execution of these steps.

Assembly results

This pipeline has been tested on an Illumina *E. coli* [42], [43] dataset. Note that current Illumina software can filter the data by removing reads that do not pass the GA analysis software called `Failed_Chastity`. To stress-test the assemblers on harder datasets,

[40] G. Narzisi. *Scoring-and-Unfolding Trimmed Tree Assembler: Algorithms for Assembling Genome Sequences Accurately and Efficiently*. PhD thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, 2011.

[41] The description below is based on joint work with Giuseppe Narzisi, also described in Narzisi's PhD thesis.

[42] Illumina. De novo assembly using illumina reads. *Technical Note: sequencing*, 2010. URL `http://www.illumina.com`.

[43] F.R. Blattner, G. Plunkett, C.A. Bloch, N.T. Perna, V. Burland, M. Riley, J. Collado-Vides, J.D. Glasner, C.K. Rode, G.F. Mayhew, J. Gregor, N.W. Davis, H.A. Kirkpatrick, M.A. Goeden, D.J. Rose, B. Mau, and Y. Shao. The complete genome sequence of escherichia coli k-12. *Science*, 277(5331):1453–1462, 1997.

in this study, we use the full output of the machine, usually contained in the `export` file. This dataset consists of 49 million 125 bp long reads, for a total coverage 1320×. Since such a high coverage is not typically available for larger genomes, it was subsampled only at100× coverage for comparing the results.

Table 1 shows a comparison of the assemblies obtained by SUTTA both on the original read set (created by Bustard) the error-corrected set (base called by TotalReCaller). SUTTA's performance significantly improves on the new reads generated by TotalReCaller. For comparison, SUTTA was tested against some of the best assemblers for short read technology on the *E. coli* dataset, specifically SOAPdenovo, ABySS and Velvet. The results are reported in table 1. Since the reads are already 125 bp long, only contigs with size ≥ 200 have been considered in the comparison. A contig is defined to be correct if it aligns to the reference genome along the whole length with at least 95% base similarity. Inspecting the results in the table it is clear that SOAPdenovo and ABySS are particularly successful in assembling long contigs, in fact their N50 statistic is the highest. However the assembly quality is inferior to SUTTA: if only correct contigs are aligned to the reference genome, the total coverage of SOAPdenovo and ABySS are respectively 66.3% and 61.9%, while SUTTA achieves a coverage $\geq 80\%$ in all instances. This improvement could be attributed to the different assembly strategies adopted: both SOAPdenovo and ABySS first create a set of contigs solely using the read sequences and only later, in a second step, extend and merge the contigs using the mate pair information; SUTTA instead assembles the contigs by concurrently optimizing mate pairs constraints and sequence quality. Another source of the difference in behavior could be found in the error-correction technique: SOAPdenovo uses the *k*-mer analysis to correct the reads but, since this process is not error-free (i.e., has false positives), it might be introducing additional errors to the set of reads. Velvet's contigs, on the other hand, are similar in size to SUTTA's but the coverage achieved with the correct contigs is only 56.9%.

More explanatory information can be gleaned from the Feature-Response curve analysis presented in figure 2. SUTTA clearly outperforms Velvet assembly in quality. These results are in accordance with the coverage analysis presented in table 1. There were difficulties in computing the Feature-Response curve for the other two assemblers, SOAPdenovo and ABySS, because their output could not be converted into AMOS format. However, based on the previous coverage in table 1, it is fair to presume that the results would not have significantly changed.

Assembler	#corr.	N50 (kbp)	Cov. all (%)	Cov. corr. (%)
SUTTA (exp.)	339	24.1	97.4	82.7
SUTTA (draft)	168	54.6	98.2	88.6
SUTTA (ref.)	154	71.7	98.2	81.3
SOAPdenovo (ctg)	245	35.7	98.4	66.3
SOAPdenovo (scaf)	106	117.6	99.3	61.9
ABySS	92	134.4	102.9	79.7
Velvet	126	54.8	98.5	56.9

Table 1: Assembly results (contigs) for *E. coli* dataset (100X 125bp reads from one lane of Genome Analyzer II). A contig is defined to be correct if it aligns to the reference genome along the whole length with at least 95% base similarity.

Haplotypes and Architecture for GWAS

Although we started our discussions with problems stemming from an over-abundance of low-quality sequence-read data, a much bigger and harder algorithmic problem resides at the heart of genomics' scientific failure: namely, the lack of well-designed sequence assembly pipelines.

As a consequence, we don't have a single well-validated reliable reference sequence; the existing reference sequences (coming from a handful of individuals - not necessarily very representative of our shared humanity) are mostly neither haplotypic nor known to be free of rearrangement errors; we don't have enough references or a large enough library exhaustively listing genetic variants/polymorphisms; nor do we have a reliable understanding of haplotype phasing, haplo-blocks or population stratification. A rather unfortunate casualty of our disregard for basic scientific soundness lies in many recent GWAS (genome-wide-association studies) that have collected huge amounts of patient samples, from which pitifully little of substance has come out, thus failing to contribute much to our biomedical understanding of complex diseases. The phenomena, widely reported and dubbed as "our missing inheritance," has flabbergasted the entire scientific community, but has not led to a consensus on how to move beyond the resulting logjam.

It has been suggested that the solutions lie in deeper coverage (allowing one to characterize rare variants in the genomes), broader sampling of cases and controls, fewer hypotheses (e.g., focus on well-targeted genomic regions: exomes, genes, genes in certain pathways, genes connected by "networks," etc.), more hypotheses (epigenomics, microRNA's, proteomics, etc.), etc.

The best place to restart would be after re-examining our notions of "causality," namely the ones connecting genotypes to phenotypes. A particularly attractive notion of *causality*, as developed by Suppes [44] and made algorithmic by my group [45], [46] is based on the notion of probability raising and a few additional axioms to handle "screening off," Yule-Simpson effects, etc. In this setting, one could characterize and algorithmically identify a group of variants that collectively, but with a large number of small individual effects, causally determine a complex phenotype. Since the process will generate a large number of "*prima-facie causal hypotheses*," it requires auxiliary steps to separate "genuine causes" from "spurious causes." Consequently, it is critical that the hypotheses can be tested against a background of null hypotheses (thus computing a reliable *p*-value), and tightly controlling the false-discovery rate. For this purpose, it is important that we have a good model of haplotypic phasing among the non-causal variants

[44] P. Suppes. *A Probabilistic Theory of Causality*. North-Holland Publishing Company, 1970.

[45] S. Kleinberg and B. Mishra. Multiple testing of causal hypotheses. *Causality in the Sciences*, Oxford University Press, 2011.

[46] S. Kleinberg and B. Mishra. The temporal logic of token causes. *Principles of Knowledge Representation and Reasoning*, KR 2010, 2010.

that have little to do with the complex traits, but "screen off" the causal variants, since they hitch-hike in the sub-populations that also carry the trait.

This argument leads us back to many fundamental questions in genomics: creating a database of a large number (about 10,000) of haplotypic genome sequences, creating data technologies for haplotypic sequences (e.g., long-range maps plus short reads), creating accurate whole-genome haplotypic sequence assembly algorithms, validating assemblers and assemblies and finally, creating the right computational architecture that can store, move and manage large amounts of data, suitably corrected and compressed. Most of the basic ingredients for these steps already exist and can be integrated without requiring a massive amount of resources.

Conclusion

Moving forward, one may wonder what a meaningful Biotechnological Moore's law should look like and what set of principles should drive it. Clearly, such a law should enable a heterodox technologically agnostic combination of a diverse set of ideas, technologies and disciplines. This Moore's law should not hinder the evolution of a complex biotechnological eco-system, which would be composed of many inter-dependent component technologies, and yet be driven by a Darwinian competition, favoring the fittest. We outline desiderata, influenced to some degree by the success of the computational Moore's law.

Miniaturization

- *Single Molecule, Single Cell, Nano-scale and Femto-second Technology*: We expect to see a trend towards usage of smaller numbers of molecules (thus avoiding errors due to loss of synchronization of basic indivisible biochemical steps), with the ultimate goal of going down to just one single molecule. Similarly, to avoid errors due to cell heterogeneity, one must aim at working with very few cells (e.g., one single cell). These trends will also motivate technologies for direct – but non-invasive – manipulation of single-molecule objects with high resolution, speed and degree of care.
- *Minute amount of material*: The technologies will also require simple and fast sample preparation, which would necessarily avoid amplification or other complex chemistry, since such steps will not be amenable to

rapid automation, but may introduce many undesirable artifacts.

- *Non-Invasive, Asynchronous and Non-Realtime*: The ideal technology must aim to be non-invasive (i.e., will not destroy the sample, as would be the case with one based on electron microscopy) and asynchronous (i.e., not requiring multiple subtasks to be synchronized). However, the amount of data should be processable and compressible with relative ease, and thus, reduce the burden on the data network, data storage and the computational architecture.

Abstraction

- *Multi-disciplinary, yet allowing Inter-disciplinary Abstraction*: The ideal technology must allow abstraction, idealization and approximation among different levels so as to permit different multi-disciplinary teams to operate at multiple levels without imposing unnecessary constraints on the others. For instance, a population genomicist working on genome-wide association studies should not have to deal with missingness imposed by the errors in base calling analysis, which in turn could depend on particular idiosyncrasies of the chemistry (e.g., homo-polymer ambiguities).

Modularity

- *Optimal Integration of Several Technologies*: For instance, a set of technologies based on manipulation of immobile single molecules on a surface or mobile single molecules in a nano-pore.
- *Order of Emphasis on Technologies*: While the combined technologies may be partly computational, partly physical, and partly chemical, an ideal mix could be 85%, 10% and 5%, respectively. While there is no hard-and-fast rule to justify such a portfolio structure, our intuition is that such a mix would allow the integrated technology to take advantage of existing Moore's laws (e.g., in CPU, communication, storage, sensing, optics, sample preparation and reagents).

Error Resilience

- *"Reliable Technologies" out of Unreliable Parts*: A hallmark of a rapidly growing technology must be its ability to withstand uncertainty, errors and even occasional

catastrophic failures in the underlying low-level components. By repeated experiments, error-correcting codes and error detection, it should be able to produce the correct results (or detect and discard incorrect results, which may happen occasionally). In these schemes, accurate modeling of the error sources and their usage through Bayesian priors will (and already do) play a critical role.

- 0-1 *Laws and Experiment Design*: Many of the genome sequencing/mapping problems contain intractable (e.g., NP-hard) problems at their core, thus suggesting a pessimistic consequence. If a well-known conjecture (viz., $P \neq NP$) is true, these problems are expected to have unreasonable worst-case complexity. However, in many cases, through well-conceived experimental designs, it is possible to engineer the system to only have to deal with "easy" instances of these hard problems. Thus, it would be imperative that the technology design exploits these "computational phase transitions," so that it does not squander its progress in throughput or cost benefits only to be defeated by a subsequent intractable computational problem.

Over the last five years or so, my laboratory has been developing a cost-effective whole-genome haplotypic sequencing technology, called S*M*A*S*H that attempts to fulfill the set of desiderata listed above. S*M*A*S*H, instead of building a monolithic technology *ab initio*, seeks to combine many design principles that have already been explored, namely in the context of optical mapping, sequencing by hybridization (SBH) and algorithms for haplotypic assembly of SBH data. The key ideas behind the technology are rather simple: Since a single-molecule technology such as optical mapping can provide accurate long-range ordered-restriction site information (over a single molecule with a length of about $500Kb$), with sufficient coverage and in conjunction with probe-hybridization data, it is not too difficult to construct a haplotypic whole-genome optical map for an individual. Such a process produces two nearly identical ordered restriction maps for each diploid autosomal chromosome – along with "k-mer spectra" for each restriction fragment in the map. If the optical map uses a 6-cutter restriction enzyme and a set of all (about 2000) 6-mer probes, modulo reverse complementation, then the 6-mer spectra of a restriction fragment (of average length of about $4Kb$) consists of all the 6-mers occurring within any window of about $400bps$. By analyzing a sliding window of about $400bps$ length, the S*M*A*S*H algorithm creates the most plausible sequence for each

of the restriction fragments: i.e., the sequence that would have created the observed spectra under a Bayesian prior, accounting for various error processes. Further optimization in the choice of restriction enzymes, and in the probe design (using "don't care" or universal bases) could be augmented to improve the technology cost-effectively.

As a first step towards this goal, we implemented S*M*A*S*H by building upon a set of simple experimental feasibility studies, from which it was possible to accurately estimate the error parameters likely to be involved in such a system. Next, by incorporating realistic parametric descriptions as Bayesian priors to the S*M*A*S*H assembler, we proceeded to validate the quality of assembly by a rigorous simulation study. From these empirical studies, it appears to be possible to obtain human-scale haplotypic whole-genome assembly with an error rate as low as $1bp/Mb$ and without any rearrangement or haplotypic ambiguities. While practical construction of a set of full-scale whole-genome haplotypic reference human sequences for single individuals is yet to be demonstrated, this technology delineates how to accomplish these goals in the not-so-distant future[47].

[48]

[47] Methods, computer-accessible medium, and systems for generating a genome wide haplotype sequence; US20080228457: Mishra, Bhubaneswar; Anantharaman, Thomas; and Lim, Sang, 2011

[48] This paper builds on my joint work with Dr. G. Narzisi of CSHL, Dr. F. Vezzi of Univ. Udine and P. Franquin and F. Menges of NYU, and directly incorporates portions of our joint publications. The paper has improved considerably following many insightful suggestions from several colleagues: most notably, A. Witzel of NYU and T.S. Anantharaman of BioNanoGenomics.

References

B. Alcott. Jevons' paradox. *Ecol. Econ.*, 54:9–21, 2005.

T.S. Anantharaman and B. Mishra. Genomics via optical mapping (i): 0-1 laws for single molecules. *Technical Note: Unpublished Manuscript*, 2000. URL `http://www.cs.nyu.edu/mishra/PUBLICATIONS/04.gvom.pdf`.

T.S. Anantharaman, V. Mysore, and B. Mishra. Fast and cheap genome wide haplotype construction via optical mapping. *The Pacific Symposium on Biocomputing*, PSB 2005:385–396, 2005.

C. Aston, D.C. Schwartz, and B. Mishra. Optical mapping and its potential for large-scale sequencing projects. *Trends in Biotechnology*, 17:297–302, 1999.

T. Berger. *Rate Distortion Theory*. Prentice-Hall, 1971.

R. Bisiani. *Beam search*, pages 56–58. Wiley & Sons, 1987.

F.R. Blattner, G. Plunkett, C.A. Bloch, N.T. Perna, V. Burland, M. Riley, J. Collado-Vides, J.D. Glasner, C.K. Rode, G.F. Mayhew, J. Gregor, N.W. Davis, H.A. Kirkpatrick, M.A. Goeden, D.J. Rose, B. Mau, and Y. Shao. The complete genome sequence of escherichia coli k-12. *Science*, 277(5331):1453–1462, 1997.

Y. Erlich, P.P. Mitra, M. delaBastide, W.R. McCombie, and G.J. Hannon. Alta-Cyclic: a self-optimizing base caller for next-generation sequencing. *Nat. Methods*, 5:679–682, Aug 2008.

A. Lim et al. Shotgun optical maps of the whole escherichia coli 0157:h7 genome. *Genome Research*, 11(9):1584–1593, 2001.

J. Lin et al. Whole genome shotgun optical mapping of deinococcus radiodurans. *Science*, 285(5433):1558–1562, 1999a.

Z. Lai et al. A shotgun sequence-ready optical map of the whole plasmodium falciparum genome. *Nature Genetics*, 23(3):309–313, 1999b.

P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *FOCS*, pages 390–398, 2000.

Illumina. De novo assembly using illumina reads. *Technical Note: sequencing*, 2010. URL `http://www.illumina.com`.

W.S. Jevons. *The Coal Question*. Macmillan and Co., 1865.

S. Kleinberg and B. Mishra. Multiple testing of causal hypotheses. *Causality in the Sciences*, Oxford University Press, 2011.

S. Kleinberg and B. Mishra. The temporal logic of token causes. *Principles of Knowledge Representation and Reasoning*, KR 2010, 2010.

A.H. Land and A.G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

E.S. Lander and M.S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2 (231), 1988.

R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, S. Li, H. Yang, J. Wang, and J. Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20(2):265–272, 2010.

F. Menges, G. Narzisi, and B. Mishra. Totalrecaller: Improved accuracy and performance via integrated alignment & base-calling. *Bioinformatics*, 5(7), 2011.

B. Mishra. Optical mapping. *Encyclopedia of the Human Genome*, 4: 448–453, 2003.

G. Narzisi. *Scoring-and-Unfolding Trimmed Tree Assembler: Algorithms for Assembling Genome Sequences Accurately and Efficiently*. PhD thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, 2011.

G. Narzisi and B. Mishra. Comparing de novo genome assembly: The long and short of it. *PLoS ONE*, 6(4):e19175, 2011a.

G. Narzisi and B. Mishra. Scoring-and-unfolding trimmed tree assembler: concepts, constructs and comparisons. *Bioinformatics*, 27(2):153–160, 2011b.

A.M. Phillippy, M.C. Schatz, and M. Pop. Genome assembly forensics: Finding the elusive misassembly. *Genome biology*, 9: R55, 2008.

J.T. Simpson, K. Wong, S.D. Jackman, J.E. Schein, S.J. Jones, and A. Birol. Abyss: A parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009.

P. Suppes. *A Probabilistic Theory of Causality*. North-Holland Publishing Company, 1970.

F. Vezzi, G. Narzisi, and B. Mishra. Feature-by-feature – evaluating de novo sequence assembly. *PLoS ONE*, page (In Press), 2012.

D.R. Zerbino and E. Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5): 821–829, 2008.