1 INTRODUCTION

Shannon information theory and biology have a long and, at times, controversial relationship (Adami, 2004; Gaftin, 1972; Quastler, 1953). As well presented by Godfrey-Smith and Sterelny (2008), that seems to be mostly due to the expectation one has about the word ‘information’ in a biological system as opposed to in a signal transmitting one. For many years, the most successful use of information theory methods has been for sequence analysis and, in particular, to measure the ‘deviation from randomness’ of nucleotide and amino acid sequences (Konopka, 2005). As Konopka points out, information theory does not seem to be essential for that task, because it could be performed by other means. Yet, given the actual deluge of data and the shift towards system-wide views of biology, it is argued that information theory may offer some advantages for data analysis over traditional statistical methods (Rissanen et al., 2007). While those issues are being debated, it is worth recalling that textual data compression, one of the quintessential contributions of information theory to science and technology, has been found to be fundamentally connected to classification, statistics and various notions of sequence complexity (e.g. Allison and Yee, 1990; Allison et al., 1992; Barron et al., 1998; Bolshoy, 2003; Cover and Thomas, 1991; Lempel and Ziv, 1976; Li and Vitányi, 1997; Rissanen and Yu, 2000; Ziv, 1988), all crucial for bioinformatics. Despite those connections, the applicability of data compression to tasks in the computational biology sciences has been somewhat underestimated, probably due to the dispersion of results over conferences and journals catering to different scientific communities. One of the major conclusions stemming from this effort is the pervasiveness of data compression in computational biology and the ubiquitousness of the associated techniques. In fact, we identify 10 areas of relevance for computational biology, in which data compression techniques have either resulted in the development of top-ranking methods or have been the fulcrum for major theoretic break-throughs, which need to be duly followed by additional work to result in valuable tools. Accordingly, the remainder of this review is organized as follows. Sections 2 and 3 report research in two of the most canonical areas of data compression: storage and entropy estimation. Sections 4-8 highlight contributions to areas of bioinformatics that are perceived as being of a fundamental nature and of broad interest and applicability, ranging from efficient support of pattern matching primitives to speed-ups of well-known dynamic programming algorithms. All of those results have their roots in ground-breaking theoretic advances with an initial fallout in terms of valuable tools for bioinformatics, although additional research is required to bring those areas to their full potential. The following three sections offer additional areas of bioinformatics where data compression and some related information-theoretic techniques have been used. They all deal with the ‘discovery or inference of structure’ in biological data, including networks. In the final section, some conclusions are drawn about the use of data compression for biological investigation.

2 COMPRESSION FOR STORAGE OF BIOLOGICAL SEQUENCES

Grumbach and Tahi (1993, 1994), in their seminal papers about the challenges of compression of DNA sequences, propose the following two scenarios for the problem:

- **Horizontal mode**: one is given a biological sequence, which is compressed by making use of information contained only in the sequence, typically by making reference only to its substrings. Evaluation of compression methods is usually performed in this mode.
• Vertical mode: one is given a set of biological sequences and each sequence is compressed by making use of information contained in the entire set, typically the substrings of the set.

The horizontal mode finds its motivation both in theory and in practice. In theory, it is of interest in order to shed light on the statistical and structural properties of biological sequences, as outlined in Sections 3, 9 and 10. In practice, it is of interest for the reduction of storage and transmission costs. The vertical mode finds the same practical motivation as the horizontal one, but it has a rather different theoretic root. In fact, one can observe, pragmatically, that, although each biological sequence may be difficult to compress, a group of related sequences, i.e. similar in function and as sequences, may compress well together. For later use, we refer to such a pragmatic observation as relative compressibility (RC) which, as will be discussed in Section 6, turns out to be a fundamental notion for classification.

2.1 Substitutional–Statistical methods
This class of methods combines the two most well-known and successful compression techniques: substitutional (Storer and Szymanski, 1982), and statistical (Cover and Thomas, 1991). An outline of those two paradigms is given in the Supplementary Material. Basically, the sequence to be compressed is partitioned into substrings, some of which are compressed well via substitutional methods, while the remaining ones are compressed well via statistical methods. A suitably defined gain function is used to establish the division of the substrings in the two groups. This paradigm has been initiated by Biocompress 1 and 2 (Grumbach and Tahi, 1993, 1994) and offers a wide variety of methods with a range of appealing choices in terms of the trade-offs between compression and speed, e.g., 3Flact (Rivals et al., 1996b), OFF-Line (Apostolico and Lonardi, 1998), GenCompress (Chen et al., 2000) and its improved version DNACompress (Chen et al., 2002), CTTW-LZ (Matsumoto et al., 2000), CASTORE (Benci et al., 2004), DNAc (Manzini and Rastero, 2005), LUT (Bao et al., 2005), DNA Pack (Belzadi and Fessant, 2005), NMLComp (Tabus et al., 2003) and the closely related methods ProtComp (Hategan and Tabus, 2004) and Getz (Korodi and Tabus, 2005). Most of those methods use the peculiar nature of redundancy in biological sequences that presents itself under the form of reverse complement matches and approximate repeats.

To the best of our knowledge, 2xI (Cao et al., 2007), is the first pure statistical compression method for biological sequences. Following that general scheme, 2xI compresses each symbol in a sequence using arithmetic coding (Witten et al., 1999) and an adaptive model for symbol probability distribution. This distribution is computed and updated via a combination of 'expert' models, where each model specializes for a particular type of statistical information in the sequence and has been carefully designed on a sound biological hypothesis. To date, based on experiments on benchmark datasets (see Supplementary Material), 2xI seems to be the compression method of choice, both on DNA and proteins, guaranteeing improvements in both compression and running time. For instance, on a DNA corpus of sequences, the average compression ratio (bits per symbol) is 1.6940 as opposed to 1.7148 achieved by DNA Pack, the best performing of the methods against which 2xI has been compared. Moreover, its performance compares favorably with the highly specialized method ProtComp for protein sequences, i.e. 3.9434 bits per symbol. In addition to its versatility in compressing biological sequences, 2xI offers the advantage of computing the information content of a sequence per base. In turn, that can be used to identify areas of interest, e.g. repeated subsequences or low complexity regions, as the authors demonstrate on the HUMHBB human gene. We anticipate that the identification of repetitions, ‘unusual’ subsequences and low complexity regions are recurring themes in the application of data compression techniques to the analysis of biological sequences. Although Sections 3.3 and 10 are specifically dedicated to those aspects of sequence analysis, most of the methods presented in this survey are relevant for those problems.

2.2 Transformational methods
The Burrows–Wheeler transform (Burrows and Wheeler, 1994) is the most well-known example in this class (see Supplementary Material), where the sequence is subject to transformations before the actual compression takes place. Based on that transform, there are only two methods, variants of each other, that specialize in biological sequences (Adjeroh and Nan, 2006; Adjeroh et al., 2002). The latest of the two has been a big step forward in protein sequence compression, yielding, also, novel insights into protein sequence structure on a genomic scale. In fact, applying their technique to several proteomes, Adjeroh and Nan (2006) provide experimental evidence that redundancy in protein sequences is in the form of repeated subsequences that are separated by thousands of symbols, e.g. 350,000 in one case for Homo Sapiens. This scale of redundancy has not been observed before, even with the use of computational methods. Although multiple gene copies and repeated histone clusters are known to be present in most eukaryotic genomes, their number and their sizes do not seem to be enough to explain such ‘long range’ correlations in protein sequences. Probably, lack of knowledge about sequence structure is the reason for the apparent incompressibility of protein sequences. On this topic, see also Nevill-Manning and Witten (1999), Hategan and Tabus (2004) and Section 3.1.

2.3 Grammar-based methods
In this class of methods, a text string $x$ is compressed by inferring or using a context-free grammar $G(x)$ to generate it. Then, the string is encoded by a proper encoding of the relevant production rules (see Supplementary Material and references therein). For biological sequences, there are three methods in this class. DNASequitur (Chemivsky and Ladner, 2004) is a straightforward extension of the Sequitur method (Nevill-Manning and Witten, 1997), where the only addition is the use of reverse complements as a source of duplication. RNACompress (Liu et al., 2008) is specific for RNA, with two main goals in mind: (i) RNA structural data compression; (ii) design of a model to represent RNA secondary structure as well as to derive its informational complexity, i.e. Kolmogorov complexity (see Section 6). For (i), experiments show that RNACompress yields an improvement in compression ratio, ranging from 5% to 50%, with respect to the reference method GenCompress, with a gain in compression/decompression speed of two orders of magnitude. As for (ii), the authors contribute a sound and general definition of information content of RNA secondary structure, giving a substantial methodological and experimental contribution to a research line initiated by Carothers et al. (2004) in
Table compression, introduced by Buchsbaum et al. better assess the ability of table compression to mine biologically for biodiversity studies. A more indepth study is under way to on a table of 1000 specimens collected over the years at InBio has been conducted, with very encouraging preliminary results, compression as a tool for biological investigations. The experiment compression methods can be successfully used for classification, compression/decompression speed. with respect to generic compression programs, with essentially the same gain in compression ratio ranging from 80% to over 150% with ratios of 100:1 or better are desirable. It has been applied with very brilliant results to different types of tabular data, including multiple alignments from the PFAM database (Buchbaum et al., 2003; Vo and Vo, 2004, 2007), where, on a selected number of alignments, the best performing table compression method yields a gain in compression ratio ranging from 80% to over 150% with respect to generic compression programs, with essentially the same compression/decompression speed.

2.4 Table compression

Table compression, introduced by Buchsbaum et al. (2000), is a unique application of compression to massive storage and transmission of data and it is another incarnation of RC, just like the vertical mode of compression for biological sequences. Its goals are to be fast, online and effective: eventual compression ratios of 100:1 or better are desirable. It has been applied with very brilliant results to different types of tabular data, including multiple alignments from the PFAM database (Buchbaum et al., 2003; Vo and Vo, 2004, 2007), where, on a selected number of alignments, the best performing table compression method yields a gain in compression ratio ranging from 80% to over 150% with respect to generic compression programs, with essentially the same compression/decompression speed.

Apostolico et al. (2008) have recently shown that table compression methods can be successfully used for classification, shading a new and important methodological light on table compression as a tool for biological investigations. The experiment has been conducted, with very encouraging preliminary results, on a table of 1000 specimens collected over the years at Ilobio for biodiversity studies. A more indepth study is under way to better assess the ability of table compression to mine biologically meaningful correlations in data.

3 ENTROPY ESTIMATORS

In information theory, entropy is a measure that allows for the evaluation of the level of ‘randomness’ in a string of symbols. Because of the duality of randomness/structure, it seems important to estimate the information content of biological sequences in order to acquire information on the ‘model’ generating them that may, in turn, shed light on structure and function, e.g. characterization/identification of coding regions, exons, introns and so on. In fact, starting with pioneering work by Schneider et al. (1986) and Getzell et al. (1992), the last decade has seen the appearance of many methods that estimate the entropy of biological sequences. They use three pillars of information theory (Cover and Thomas, 1991): (A) the asymptotic equipartition property (AEP), discussed in Section 3.1; (B) universality theorems, discussed in Section 3.2 and (C) Rényi entropy, discussed in Section 3.3.

3.1 Methods based on the AEP

Let be an alphabet of symbols, let be the probability of the -th string in in lexicographic order and let be the entropy of the information source . The AEP (see Supplementary Material) reduces the problem of estimating the entropy of a source to that of estimating , for large enough . However, a direct estimate of would suffer from the ‘finite sample effect’, as the value of grows, only a small fraction of the possible strings will appear in the sample, resulting in a poor approximation of the joint probability distribution and therefore of the entropy.

Methods in this class that overcome the mentioned problem are reported in (Li et al. 1996), (Schmidt and Weiss 1998), Crochemore and Vérin (1999), Loewenstern and Yianilos (1999), Weiss et al. (2000) and Benedetto et al. (2007). Most of those studies have been directed at investigating the level of randomness in biological sequences, yielding a wealth of results about their informational structure. Among those, we limit ourselves to mention only a few results. It has been shown that protein sequences seem to be fairly random, although medium- and long-range correlations among amino acids are present and responsible for some redundancy. This is consistent with analogous findings obtained in studying the compressibility of protein sequences, mentioned in Section 2.2. As for DNA sequences, experiments conducted on the Epstein Bar virus show that, when compared with other textual information carriers, e.g. text or computer code, they have greater freedom in combining alphabet symbols; that is, they look like random sequences. Moreover, studies conducted on whole chromosomes of S. cerevisiae and large parts of the E. coli genome show that there is a bulk of homogeneity at the chromosomal or genomic level, although the statistical properties of DNA are largely locally inhomogeneous. It seems that biases in mutational pressure and recombination processes are responsible for the homogenization process.

3.2 Methods based on universality theorems

A universality theorem for a given data compression algorithm C is a very powerful mathematical statement about C, with remarkable practical implications (see Supplementary Material). Informally, it states that, given a long enough string, the compression ratio achieved by C tends to . All that is without any knowledge of the statistical properties of the source, which are ‘learned’ by the algorithm. So, any universal data compressor can be seen as an entropy estimator. It is unfortunate that many of the methods presented in Section 2 are of little use as entropy estimators of biological sequences: the convergence of the compression rate to the entropy of the source is too slow.

Two methods are known to overcome this ‘slow convergence’ problem. The one by Farach et al. (1995) is a non-trivial variation of the Ziv and Lempel (1977, 1978) compression algorithms. With its use, two tests have been performed in order to assess how the entropy of exons compares with that of introns in human sequences, Weiss and Herzel (1998), Crochemore and Vérin (1999), Loewenstern and Yianilos (1999), Weiss et al. (2000) and Benedetto et al. (2007). Met in Section 3.2 and (C) Rényi entropy, discussed in Section 3.3.

1577
of exons is higher than that of introns, a somewhat surprising result because introns are presumed to be the mechanism by which many random changes can accumulate without being subjected to restorative survival forces. The method by Lanctot et al. (2000) is strongly related to grammar-based compression methods, with a few major variations. It is a very fast method, giving excellent entropy estimates: on benchmark data (see Supplementary Material), this method gives an average estimate of 1.66 bits per symbol as opposed to the one of 1.71 obtained by the reference algorithm of Loewenstern and Yianilos (1999). It has been used to measure the entropy of coding and non-coding regions in E.coli and it has been found that non-coding regions have a lower entropy than coding regions, which agrees with results by Farach et al. (1995). Moreover, the method has also been used to measure the difference in entropy between highly expressed essential genes and ‘normal’ genes in order to test the hypothesis that random mutations in ‘normal’ genes in regions, which agrees with results by Farach found that non-coding regions have a lower entropy than coding regions, which agrees with results by Farach et al. (1995). Moreover, the method has also been used to measure the difference in entropy between highly expressed essential genes and ‘normal’ genes in order to test the hypothesis that random mutations in ‘normal’ genes are less likely to be deleterious than in highly expressed essential genes. It turned out that highly expressed genes have a lower entropy estimate than ‘normal’ genes and that, using statistical tests, the mentioned hypothesis is supported with over 99% confidence.

3.3 Methods based on Rényi entropy

Rényi entropy (Rényi, 1961) is a generalization of continuous functions of Shannon entropy (see Supplementary Material). It has been used, primarily, for pattern and motif discovery in biological sequences, with particular attention to the identification of binding sites and other regulatory regions (see also Section 10). Schneider et al. (1986) developed a method that uses Shannon entropy function for the identification of binding sites. It was validated experimentally on E.coli. That method is based on the finding that redundancy is close to zero in subsequences ‘surrounding’ a binding site and it is substantially greater than zero for the subsequence ‘containing’ the binding site. That is, binding sites have more structure with respect to subsequences ‘around’ them. Such a change in redundancy values highlights good binding sites. Following that line of research, Krishnamachari et al. (2004) proposed the use of the discrete version of Rényi entropy for the same problem, showing that the latter is better than the former, in particular with respect to range identification, i.e. the length of the binding site. Moreover, a particular incarnation of the Rényi continuous entropy function has been proposed by Vinga and Almeida (2004) for the estimation of the complexity of biological sequences and later applied (Vinga and Almeida, 2007) to compute entropic profiles, e.g. graphs of information content per base, of DNA sequences with the aim of finding ‘unusual’ regions that may also turn out to be biologically relevant. The authors have applied their entropic profiling method to both E.coli and Haemophilus influenzae genomes, reporting that it correctly identified known regulatory components and motifs, both in regard to position and scale (length) of conserved segments.

4 SPACE-TIME-EFFICIENT, GENOME-WIDE, STRING MATCHING PRIMITIVES

Suffix trees and arrays have come of age in bioinformatics (Gusfield, 2002) where, thanks to their ability to support, efficiently, a variety of exact string matching and word counting operations (Apostolico, 1985), they now make a difference in a wide range of bioinformatics applications. While this process took place, the computer science literature witnessed a major breakthrough: the remarkable discovery of self-indexes, i.e. data structures analogous to suffix trees and arrays, but with space requirements theoretically close to the entropy of the sequences to be indexed, with no substantial slow-down in search time, and able to reconstruct any portion of the sequences on demand (with the implication that the sequences no longer need to be stored separately). The state of the art is well presented in Navarro and Mäkinen (2007) and Ferragina et al. (2008).

Since high memory demand is a major bottleneck for the application of suffix trees and arrays on a genomic scale, the use of self-indexes in bioinformatics has been immediately investigated, initially by Sadakane and Shibuya (2001). The first convincing use of compressed suffix arrays (CSAs) for genomic research was given by Healy et al. (2003) that, motivated by oligonucleotide probe design, implemented a version of the CSA able to store the forward sequence of the human genome in 1G of main memory.

They also demonstrated how efficiently one can process simple string matching queries. For instance, annotating with counts all overlapping words of length 24 in the human genome could be done at a speed of 1 min/MB on a PC.

The study also showed that, with the use of CSAs, one can perform string matching tasks on a genomic scale, e.g. the identification of large repeats, that could not be possible with the use of other string matching data structures. A major drawback of this approach is that the spacecraft required for the construction of the CSA was still large. In fact, it was built with the use of a cluster of 16 processors. Lippert et al. (2005) have contributed to that ground-breaking work with a demonstration of the ground-breaking implementation of a CSA that could be built, on the human genome, on a workstation in <2G of workspace, removing the ‘big-memory’ computation step from large genome exact matching problems. Moreover, Lippert (2005) (see also errata at the author’s home page) showed how to use the new version of the CSA for space-efficient whole-genome sequence comparison. In particular, he showed that all 20mers in common between the human and the mouse genomes could be computed in a couple of days on a PC, while the best implementations of suffix trees and arrays would take at least twice that time. Another demonstration of the genome-wide possible use of self-indexes is given by Válimäki et al. (2007), where a new type of self-index, called self-indexes in bioinformatics has been immediately investigated, initially by Sadakane and Shibuya (2001). The first convincing use of compressed suffix arrays (CSAs) for genomic research was given by Healy et al. (2003) that, motivated by oligonucleotide probe design, implemented a version of the CSA able to store the forward sequence of the human genome in 1G of main memory.

They also demonstrated how efficiently one can process simple string matching queries. For instance, annotating with counts all overlapping words of length 24 in the human genome could be done at a speed of 1 min/MB on a PC. The study also showed that, with the use of CSAs, one can perform string matching tasks on a genomic scale, e.g. the identification of large repeats, that could not be possible with the use of other string matching data structures. A major drawback of this approach is that the spacecraft required for the construction of the CSA was still large. In fact, it was built with the use of a cluster of 16 processors. Lippert et al. (2005) have contributed to that ground-breaking work with the design and implementation of a CSA that could be built, on the human genome, on a workstation in <2G of workspace, removing the ‘big-memory’ computation step from large genome exact matching problems. Moreover, Lippert (2005) (see also errata at the author’s home page) showed how to use the new version of the CSA for space-efficient whole-genome sequence comparison. In particular, he showed that all 20mers in common between the human and the mouse genomes could be computed in a couple of days on a PC, while the best implementations of suffix trees and arrays would take at least twice that time. Another demonstration of the genome-wide possible use of self-indexes is given by Válimäki et al. (2007), where a new type of self-index, called self-indexes in bioinformatics has been immediately investigated, initially by Sadakane and Shibuya (2001). The first convincing use of compressed suffix arrays (CSAs) for genomic research was given by Healy et al. (2003) that, motivated by oligonucleotide probe design, implemented a version of the CSA able to store the forward sequence of the human genome in 1G of main memory.

They also demonstrated how efficiently one can process simple string matching queries. For instance, annotating with counts all overlapping words of length 24 in the human genome could be done at a speed of 1 min/MB on a PC. The study also showed that, with the use of CSAs, one can perform string matching tasks on a genomic scale, e.g. the identification of large repeats, that could not be possible with the use of other string matching data structures. A major drawback of this approach is that the spacecraft required for the construction of the CSA was still large. In fact, it was built with the use of a cluster of 16 processors. Lippert et al. (2005) have contributed to that ground-breaking work with the design and implementation of a CSA that could be built, on the human genome, on a workstation in <2G of workspace, removing the ‘big-memory’ computation step from large genome exact matching problems. Moreover, Lippert (2005) (see also errata at the author’s home page) showed how to use the new version of the CSA for space-efficient whole-genome sequence comparison. In particular, he showed that all 20mers in common between the human and the mouse genomes could be computed in a couple of days on a PC, while the best implementations of suffix trees and arrays would take at least twice that time. Another demonstration of the genome-wide possible use of self-indexes is given by Válimäki et al. (2007), where a new type of self-index, called self-indexes in bioinformatics has been immediately investigated, initially by Sadakane and Shibuya (2001). The first convincing use of compressed suffix arrays (CSAs) for genomic research was given by Healy et al. (2003) that, motivated by oligonucleotide probe design, implemented a version of the CSA able to store the forward sequence of the human genome in 1G of main memory.

They also demonstrated how efficiently one can process simple string matching queries. For instance, annotating with counts all overlapping words of length 24 in the human genome could be done at a speed of 1 min/MB on a PC. The study also showed that, with the use of CSAs, one can perform string matching tasks on a genomic scale, e.g. the identification of large repeats, that could not be possible with the use of other string matching data structures. A major drawback of this approach is that the spacecraft required for the construction of the CSA was still large. In fact, it was built with the use of a cluster of 16 processors. Lippert et al. (2005) have contributed to that ground-breaking work with the design and implementation of a CSA that could be built, on the human genome, on a workstation in <2G of workspace, removing the ‘big-memory’ computation step from large genome exact matching problems. Moreover, Lippert (2005) (see also errata at the author’s home page) showed how to use the new version of the CSA for space-efficient whole-genome sequence comparison. In particular, he showed that all 20mers in common between the human and the mouse genomes could be computed in a couple of days on a PC, while the best implementations of suffix trees and arrays would take at least twice that time. Another demonstration of the genome-wide possible use of self-indexes is given by Válimäki et al. (2007), where a new type of self-index, called self-indexes in bioinformatics has been immediately investigated, initially by Sadakane and Shibuya (2001). The first convincing use of compressed suffix arrays (CSAs) for genomic research was given by Healy et al. (2003) that, motivated by oligonucleotide probe design, implemented a version of the CSA able to store the forward sequence of the human genome in 1G of main memory.
Xiong (2001). However, its best use seems to be for classification compression of biological sequences and table compression, as be established via RC, which also supports the vertical mode of related and vice versa. Therefore, a similarity of sequences can USM must be approximated, usually approximating Kolmogorov complexities are non-computable in the Turing sense, of y approximations of USM depend, critically, on which data compressors are used. For instance, all three approximations of USM depend, critically, on C. Therefore, RC and USM are methodologies used to compute similarity between sequences, rather than being formulae or procedures returning a numeric value. Before addressing performance issues, we present two domains of computational biology where the methodologies have been applied or their potential profitable application has been discussed.

Phylogeny (Apostolico et al., 2006; Cilibrasi and Vitányi, 2005; Ferragina et al., 2007; Li et al., 2001, 2003; Otu and Sayood, 2003b; Rivals et al., 1996a; Ulitsky et al., 2006): Those studies use RC and USM in order to build phylogenies from entire genomes and proteomes. Classification of Proteins (Ferragina et al., 2007; Gilbert et al., 2007; Kocso et al., 2005; Krasnogor and Pelta, 2004; Liu and Wang, 2008; Pelta et al., 2005; Rocha et al., 2006): Those studies apply RC, USM and related techniques to obtain structural and evolutionary classifications of proteins, using different representations such as FASTA format files and TOPE strings.

All of those studies clearly indicate that RC and USM are worth using, even on datasets of size small enough to be processed by standard methods, including the ones based on alignments. Of particular relevance are the following facts: (i) synergies between compression-based methods and alignment methods result in superior protein classification performance with respect to HMMs (Kocso et al., 2005); (ii) among the three approximations of USM, UCD or its equivalent NCD, is worth using, since the third one is lagging behind (Ferragina et al., 2007); (iii) FPPs (Shkaran, 2002) and Genecompress are the best performers with UCD, among a broad range of compression programs used in the experimentation of Ferragina et al. (2007); (iv) reliable phylogenetic trees can be built using entire genomes and proteomes (Ulitsky et al., 2006).

We also report that Galas et al. (2008) have proposed a class of measures to quantify the contextual nature of information in sets of objects, in order to obtain a useful mathematical characterization of ‘biological information’. Once again, Kolmogorov complexity is at the heart of the theoretic foundation of those new measures. Their approximation is also investigated via NCD and data compression. Initial experiments, performed on deciphering gene interactions, show that the new measures may be of great value in biology.

Additional results as well as domains of application in biology, related to the topic of this section, can be found in Loewenstein et al. (1995), Varrel et al. (1999) and Otu and Sayood (2003a).

7 CLUSTERING AND INDEXING OF MICROARRAYS

Classification and clustering of microarray data is one of the fundamental areas of bioinformatics (Hand et al., 2005). Although the use of information theoretic concepts, such as mutual information, is not new in the design of clustering algorithms, there start to appear contributions to this area specifically designed for clustering of microarray data. Nykter et al. (2005) provide a fairly immediate extension of the similarity functions described in the previous section to microarrays. They are then applied within clustering algorithms, with some success on microarray data. Zhou et al. (2004) devise novel correlation functions among genes that are based on mutual information. Then, clustering is formulated as an optimization problem in which a suitably defined cost function is to be minimized. Particular attention is given to the methods
that evaluate the mutual information between genes. The resulting algorithm is validated on both synthetic and real microarray data. The experiments show that it substantially outperforms many classic methods. The problem of feature selection via mutual information is addressed in Long and Ding (2005) and Zhou et al. (2007). It also worth of mention that Wang, H. et al. (2002) have devised a suffix tree-based method for similarity searching in microarray databases, again with encouraging initial results.

8 SPEED-UPS OF DYNAMIC PROGRAMMING RECURRENCES: ALIGNMENTS AND HMMS

Due to their ubiquitous nature, HMMs (Durbin et al., 1999) and sequence alignment algorithms (see again, Gusfield, 1997; Kruskal and Sankoff, 1983; Waterman, 1995) play a central role in computational molecular biology. Most of the basic algorithms used by both alignment methods and HMMs are based on dynamic programming and require a superlinear running time in the input parameters, in the worst case. Therefore, they are perceived as inadequate for the analysis of long sequences where one usually resorts to heuristic algorithms. For instance, when one can relax accuracy requirements, the time-honored Smith–Waterman local alignment method (Smith and Waterman, 1981) is used after a full-fledged BLAST (Altschul et al., 1990) search has been done, in order to have a fast screening of interesting ‘similarities’. Analogously, the well-known Viterbi algorithm for HMMs (Viterbi, 1967) is rarely used on large HMMs and one resorts to various heuristic approximations in order to speed-up the computation (Buchshaus and Giancarlo, 1997). It is quite surprising, and of great theoretic relevance and potential practical impact, that the fundamental dynamic programming recurrences for alignments and for HMMs, e.g. Forward-Backward and Viterbi, can speed-up with the use of compression techniques. The speed-up for alignments is due to Crochemore et al. (2003), and despite the theoretic interest, its practicality has not been investigated. The speed-up for the HMMs recurrences is due to Mozes et al. (2007) and Lifshits et al. (2008), and to the best of our knowledge, it is the first asymptotic speed-up for this class of recurrences. Moreover, a proof of principle has been given that it is indeed practical by applying it to the CpG island identification problem (Bird, 1987), where time improvements of at least a factor of five were reported with respect to the straightforward implementation of the Viterbi algorithm. Another potential advantage of those new methods is their high degree of parallelization, as opposed to the original algorithms. Unfortunately, no systematic investigation about the algorithm engineering of this new class of methods, both on parallel and conventional computers, has been done.

9 SEGMENTATION OF BIOLOGICAL SEQUENCES

It is well known that, although DNA is very heterogeneous, there are highly homogenous regions, e.g. regions with high concentrations of G or C bases, CpG islands, ALU, LINE, low complexity repeats, etc. In order to capture important functional information, it is desirable to partition a DNA sequence into homogenous segments. Depending on the type of data and the biological information being sought, one obtains different mathematical formulations of the problem, characterizing the partition of interest via a definition of ‘homogeneity’. We briefly present two application domains where partitioning techniques have been designed, based on methodologies of interest for this review. It also worth pointing out that segmentation of sequences is a problem of broad interest and with deep connection to combinatorial optimization. The interested reader can find additional material in Hyvonen et al. (2007). Moreover, many of those approaches are based on a well-studied dynamic programming recurrence that lends itself to very efficient algorithmic solutions (Giancarlo, 1997).

9.1 Single nucleotide polymorphism and identification of haplotype blocks

Common genetic variations in human DNA sequences explain almost the entire observed differences in the phenotype of the human population, including predisposition for specific diseases. Particularly important are single nucleotide polymorphisms (SNPs) and the division of sets of haplotypes into blocks (Gabriel et al., 2002; Patil et al., 2001). A haplotype is a sequence of SNPs on chromosome that are statistically associated. A block is characterized by SNPs in close proximity, highly correlated and not easily separated by recombination. In formal terms, the identification of haplotype blocks requires the partition of a set of sequences into blocks, where the homogeneity within a block is measured by appropriate cost functions. The many computational methods available for this problem can be classified into two broad categories. In the first category, haplotype blocks are identified (via their boundaries) on the basis of the decay of Linkage-Disequilibrium (Daly et al., 2001). Methods on the second category identify blocks on the basis of some haplotype diversity measure within the blocks. Following the groundbreaking results of Zhang et al. (2002), they all have in common a dynamic programming formulation of the problem. In order to obtain such a formulation and the relevant features of it, such as the cost function assessing the homogeneity of a block, the methods by Greenspan and Geiger (2003), Koivisto (2003) and Bockhorst and Jojic (2007) make essential use of the minimum description length principle (MDL) (Barron et al., 1998).

The method by Anderson and Novembre (2003) (AN) is much more sophisticated than the ones we have mentioned because it combines both classes by making use of information on both Linkage-Disequilibrium decay between blocks and haplotype diversity within blocks. Again, the MDL principle plays a fundamental role in the development of the method, with experiments showing that it has an excellent performance with respect to other existing methods, both on real and simulated data. When applied to the data studied by Daly et al. (2001), AN finds more block boundaries in agreement with those found by Zhang et al. (2002). When applied to data simulated from the coalescent with recombination hotspots, it reliably places block boundaries at the hotspots and rarely at sites with background levels of recombination. The other three mentioned methods, on the same dataset, are either insensitive to recombination hotspots or they are not able to discriminate between background sites of recombination and hotspots. Moreover, a dataset of 822 biallelic sites in 86 complete human mtDNA sequences were used as ‘negative control’ since there is very little evidence for widespread recombination in human mtDNA, few blocks are expected to be present in the data. Again, AN found only four blocks in the data, as opposed to a considerably larger number found by the other two...
The method by Menconi and Marangoni (2006) was in the range of haplotype blocks. Those automatic tools are likely to be very useful in improving the population and genome-level processes that give rise to observations of haplotype block structure.

9.2 Change point analysis of DNA sequences and coding regions identifications

Change point analysis (also known as DNA segmentation) consists of identifying points in a DNA sequence where there is a change in homogeneity. The use of entropy and compression is not novel to this problem (e.g. Bernaola-Galván et al., 1996, 1999, 2000), although with many limitations. Szpankowski et al. (2003) have proposed a novel strategy that takes care of many of those limitations and offers a rigorous mathematical treatment of the problem with the added value of providing a clear-cut stopping rule for the algorithm that must identify the change points. In particular, the discriminant function for testing for homogeneity and block lengths has been designed using rigorous methods of information theory, i.e. universal data compression and empirically observed statistics (Ziv, 1988), in addition to the MDL. The discriminative power of the method has been assessed with the use of subsequences of human chromosomes 9 and 20. They have been chosen on the base of already available information about the starting positions of genes, coding and non-coding regions and CpG islands. The experimental evaluation has given excellent results: the identified change points are in close proximity to known boundaries between coding and non-coding regions and the start of known CpG islands.

The identification of coding/non-coding regions in DNA can be seen as a very specific segmentation task. Again, machine learning methods and HMMs are widely applied in this area (Menconi and Marangoni, 2006). However, one of their major drawbacks is the need to estimate a large number of parameters before they can actually be used. Based on CASTEORE, Menconi and Marangoni (2006) proposed a new parameter-free method, which also uses a novel measure of the information content in a sequence. Again, the input sequence is divided into blocks and changes in the information content of each contiguous block are identified. Particularly important are blocks where the information content grows sublinearly with block length, indicating the presence of regularities in the input sequence. That information, in turn, is used to discern between coding and non-coding regions. An added benefit of the method is the acquisition of a dictionary of words that collects potentially useful biological information about the sequence. Experimental results by the authors, conducted on prokaryotic genomes, indicate that the method is quite promising. It was compared against three reference, highly tuned, methods: GLIMMER, GeneMark and ZCURVE. The performance of those methods was in a range of 96–99% in prediction accuracy of annotated genes in the prokaryotic genomes used for the test. The method by Menconi and Marangoni (2006) was in the range of 88–96.6% in prediction accuracy, although it was not subject to particular optimizations and it is totally parameter-free. We also mention that a closely related approach has been used by Menconi (2004) in a prior study directed at identifying atypical regions in DNA sequences. The method was used to study 12 complete genomes of some Archaea, Bacteria and Eukaryotes, together with chromosomes 2 and 4 of Arabidopsis thaliana. Among the many areas of potential biological interest that the method highlighted in those genomes, we limit ourselves mentioning that four putative genes were identified on chromosome 2 of A. thaliana.

An independent cross-validation analysis conducted with the use of FGENESH (a HMM-based program) confirmed this finding, with the additional use of information about known positions of genes in A. thaliana.

9.3 Comparison of segmentations

There are many algorithms that find segmentations in sequences, each based on a particular set of features deemed ‘relevant’. In this context, it is essential to have techniques that compare segmentations in order to establish their relative merits. Hainmnen et al. (2007) have designed one such technique that cleverly reduces the ‘quality evaluation process’ of a segmentation to its statistical significance with respect to a background segmentation. Essential for this reduction to work is the introduction of a similarity measure between two segmentations that is based on Shannon entropy.

10 PATTERN DISCOVERY

The quest for automatic tools capable of identifying biologically relevant patterns in biosamples has resulted in the birth of a new area: pattern discovery in bioinformatics (Parida, 2007). The aim of this section is to show how data compression techniques and the associated MDL principle are used in order to discover potentially meaningful biological patterns. It is worth pointing out that other techniques presented in this review also deal with the problem of ‘discovering’ biological structure, e.g. Section 3, and in fact there is a non-trivial overlap of fundamental ideas between the methods presented here and in other sections.

10.1 Evaluating the statistical significance of patterns

The relative abundance or scarcity of occurrences of a particular subsequence in a DNA sequence seems to be a good indication of its involvement in important biological processes, such as gene regulation and DNA repair. An excellent review on this topic is provided by Reinet et al. (2005). Therefore, many research efforts have been dedicated to the assessment of the statistical significance of the occurrence of a pattern sequence in a (longer) text sequence. This scenario gives rise to two main types of problems, which we will discuss next.

The first type of problem asks for the identification of subsequences in a sequence that are statistically relevant, as established by a given measure. In this setting, Milosavljevic and Jurka (1993) and Milosavljevic (1995) have contributed ground-breaking work with the introduction of the notion of algorithmic significance in sequences, that has been further enhanced by Powell et al. (1998).

More recently, Aktulga et al. (2007) have also introduced a measure of statistical significance between sequences that can be thought of as being a variant of mutual information. The practicability and generality of this method has been assessed in two different studies, that we briefly describe. The first study was performed on the maize zm- SRp32 gene. This gene belongs to a group of genes...
that are functionally homologous to the human ASI/SF2 alternative splicing factor. Interestingly, these genes encode alternative splicing factors in maize and yet themselves are also alternatively spliced. In order to discover the amount of correlation between different parts of this gene, the mutual information was computed between all of its functional elements including exons, introns and the 5′ untranslated region. Significant dependencies were found between the 5′ untranslated region in zm-SRp32 and its alternatively spliced exons, indicating the presence of as yet unknown alternative splicing mechanisms or structural scaffolds. The second study tested the ability of the method to identify short tandem repeats in genetic profiling. Experiments conducted on the FBI’s combined index system (CODIS) show that the new method is very well suited to the task, offering good precision and a linear running time—at least definite theoretical-advantage over extant methods. On this topic, see also next the section.

The second type of problem is concerned with the extraction of significant motifs, usually represented by regular expressions, from a set of sequences. In their survey of the area, Ferreira and Azevedo (2007) suggest a division of those methods into three classes. The one termed Theoretic-Information is of relevance for this review. Brázma et al. (1996) are the first to propose a significance measure for motifs that is based on the MDL and they apply it to the Pratt pattern discovery algorithm (Jonassen, 1997). Nevill-Manning et al. (1997) propose a measure that is based both on statistics and the MDL to rank, by statistical relevance, PROSITE-like motifs. That measure is the ranking function for motifs in EMOTIFS, a pattern discovery tool also proposed by the authors. The predictive performance of EMOTIF was evaluated against a large corpus of manually derived PROSITE motifs, using a test set of sequences discovered after the PROSITE motifs were formed. In these tests, EMOTIF demonstrates vastly increased accuracy with only a comparatively small decrease in sensitivity. More recently, Ma and Wang (2000) have proposed yet another MDL-based statistical ranking function for motifs, but this time specific to the pattern discovery tool Sdiscover (Wang et al., 1994). Unfortunately, no assessment of the method is reported.

10.2 Approximate and tandem repeats

Molecular duplication mechanisms, e.g. retrotransposition, copying of genes, tandem duplication events, etc., are responsible for the presence of duplicated sequences in DNA, e.g. retroposons, microsatellites, tandem repeats, etc. The duplicated structures that those mechanisms produce perform important functions at both the regulatory and the evolutionary level. Moreover, some of them are also involved in human disease, (e.g. Madsen et al., 2008). Therefore, the identification of repeated subsequences in DNA is important and, fortunately, it is also a branch of combinatorics and algorithms with a wealth of results (Gusfield, 1997).

Data compression algorithms are natural candidates for the task of identifying repetitive areas of DNA because they exploit the presence of repeated subsequences in a sequence. Rivals et al. (1997a, b) have initiated this type of research and have investigated various aspects relating compression to the identification of repeated structures in biosquences. ARM, developed at Monash University (Allison et al., 1998; Dix et al., 2007), is a particularly sophisticated system, where algorithm engineering is complemented by a graphic ‘navigation system’ that allows for close scrutiny of the results. The techniques supporting ARM have been successfully applied to identifying both long and short repeated patterns in genomic DNA of Plasmodium Falciparum (Stern et al., 2001), leading to the hypothesis that those regions may be related to large-scale chromosomal organization and the control of gene expression. Moreover, precursors of those techniques have been applied to establishing a method for the computation of the ‘complexity’ of DNA sequences (Allison et al., 1992).

10.3 MicroRNA target detection

MicroRNAs (miRNAs) are involved in many important biological processes, e.g. gene expression regulation and silencing. Therefore, a substantial part of biomolecular research is dedicated to their study (Nature-Review, 2008) and, in particular, to the identification of their target sites.

As discussed in Evans et al. (2007), current computational methods for miRNA target site detection seem to have limitations in their specificity, returning a large number of candidate miRNA target sites. They propose a method, based on data compression and the MDL principle, that initial studies indicate is capable of identifying motif sequences, some of which turn out to be miRNA target sites involved in breast cancer. In terms of data compression techniques, the method is based on grammar inference. It can be seen as a combination of DNAsequitur and OFF-Line as well as highly engineered improvement of both.

11 COMPARISON AND INFERENCE OF BIOLOGICAL NETWORKS

The comparison of existing biological networks (Sharan and Ideker, 2006; Zhang et al., 2008) and ‘reverse engineering’ of biological networks from data on a genomic scale, i.e. gene regulatory networks from expression data (Margolin et al., 2006a), are fundamental tasks for systems biology. In fact, several research efforts are under way to tackle the computational problems associated with those tasks, although only a handful of methods are currently available. Even at such an initial stage, data compression and information-theoretic approaches are playing a fundamental role.

The network inference algorithms of relevance for this review are the ones based on mutual information, which have been used mainly for regulatory network inference, although they may also work in other contexts. We mention RELNET (Butte and Kohane, 1999, 2000), CLR (Butte et al., 2000), ARACNE (Margolin et al., 2004, 2006a), and MERLOT (Meyer et al., 2007). The basic idea is very simple and common to all of them. Given a set of elements (nodes), one builds a complete, edge-weighted graph on those nodes, where the weight on each edge gives the amount of relatedness of the two nodes, as measured by their mutual information. Then, edges with zero or low weight are removed to obtain the ‘reverse engineered’ network. Key issues for the successful application of this basic idea are (i) the accurate evaluation of the mutual information between items, which must be inferred from empirical data; (ii) filtering out false positives, in particular false direct interactions: if x interacts with y and z, while there is no direct interaction between y and z, mutual information may falsely indicate a direct interaction between y and z. Thanks to the care with which those two points have been dealt, and based on the extensive experimental studies conducted for its validation (Basso et al., 2003; Hartemink, 2005; Margolin et al.,
2006a), ARACHNE is found to be a very versatile and reliable method and therefore an entire protocol for its use in reverse engineering of cellular networks has been proposed (Margolin et al., 2006b).

As for network comparison, we are witnessing a development analogous to the one for sequence comparison. The vast majority of methods are based on notions of similarity related to extensions of alignments to graphs (Sharan and Ideker, 2006). One method, however, can rightfully be called the first to be alignment free in this novel category of algorithms (Chor and Tuller, 2007). The main ideas supporting the definition of similarity in that method, are strongly related to the ones of Section 6, but are formalized via an ad hoc use of the MDL. In fact, the proposed measure of similarity between two graphs is based on the length of the description of one graph, once the other is known. The method has been extensively tested. A first set of experiments has been conducted on the metabolic networks in the KEGG database and, based on them, phylogenetic trees for two sets of species have been built and compared with the NCBI taxonomy, showing a very good level of agreement. It is worth pointing out that the new similarity function between graphs gives rise to the only known method capable of building phylogenetic trees from network data. A second set of experiments was conducted on protein interaction networks, namely those of Drosophila melanogaster and S. cerevisiae, in order to find conserved parts. An indepth analysis of the conserved networks found by the method, via knowledge already available, indicates that it is suitable for the analysis of protein interaction networks.

12 CONCLUSIONS

Data compression, and the related information-theoretic techniques, find a wide use for investigation in computational biology. Such a pervasive use has grounds in some outstanding notions that deeply characterizes data compression, in particular universality and quantification of statistical dependence via information measures. Those notions give rise to methods that need very few assumptions on the data models and, as a consequence, very minor parameter estimations for the application of those tools. That seems to be a major advantage for computational biology applications, where the statistical modeling of the data is a highly non-trivial task. In addition, the low-computational demand of those tools allows them to scale well with dataset size, even on a genomic scale. In conclusion, versatility, ‘parameter-free’ data and association mining, and speed are the main advantages for the use of data compression in biological investigations. However, a non-trivial organizational effort is required in order for this area to collect, in a homogeneous way, the set of ideas and tools that would constitute the critical mass required to be recognized as one of the pillars in Bioinformatics. Moreover, the connection of data compression to machine learning is also receiving attention (Scutulli and Brodley, 2006) and hopefully it will result in further unifying principles and methodologies, with impact on many disciplines, including the ones connected to the Life Sciences.

ACKNOWLEDGEMENTS

The authors would like to thank Valentina Greco for contributions during the initial steps of this rather long journey. They are also grateful to Michael Levitt, Stefano Lonardi, Giovanni Manzini, Giulia Mencioni and Jacob Ziv for feedback on a draft of the review. Moreover, we are deeply indebted to Paolo Ferragina, Margaret Gagie, Travis Gagie and Michal Ziv-Ukelson for many suggestions that greatly helped the presentation of our ideas. Finally, we would like to thank Anna Tramontano, Jonathan Wren and the reviewers for a very smooth editorial process that lead to the publication of this review.

Funding: MIUR-FIRB International Project ‘Scoperta di Pattern in Struttura Discrete, con Applicazioni alla Bioinformatica’.

Conflict of Interest: none declared.

REFERENCES


Dinh, T.T. et al. (2007) Comparative analysis of long DNA sequences by per element information content using different contexts. BMC Bioinformatics, 8, s8.


