Hindawi Publishing Corporation
EURASIP Journal on Bioinformatics and Systems Biology
Volume 2007, Article ID 43670, 16 pages
doi:10.1155/2007/43670

Research Article

MicroRNA Target Detection and Analysis for Genes Related to Breast Cancer Using MDLcompress

Scott C. Evans,[1] Antonis Kourtidis,[2] T. Stephen Markham,[1] Jonathan Miller,[3] Douglas S. Conklin,[2] and Andrew S. Torres[1]

[1] *GE Global Research, One Research Circle, Niskayuna, NY 12309, USA*
[2] *Gen*NY*Sis Center for Excellence in Cancer Genomics, University at Albany, State University of New York, One Discovery Drive, Rensselaer, NY 12144, USA*
[3] *Human Genome Sequencing Center, Baylor College of Medicine, One Baylor Plaza, Houston, TX 77030, USA*

Received 1 March 2007; Revised 12 June 2007; Accepted 23 June 2007

Recommended by Peter Grünwald

We describe initial results of miRNA sequence analysis with the optimal symbol compression ratio (OSCR) algorithm and recast this grammar inference algorithm as an improved minimum description length (MDL) learning tool: *MDLcompress*. We apply this tool to explore the relationship between miRNAs, single nucleotide polymorphisms (SNPs), and breast cancer. Our new algorithm outperforms other grammar-based coding methods, such as DNA Sequitur, while retaining a two-part code that highlights biologically significant phrases. The deep recursion of *MDLcompress*, together with its explicit two-part coding, enables it to identify biologically meaningful sequence without needlessly restrictive priors. The ability to quantify cost in bits for phrases in the MDL model allows prediction of regions where SNPs may have the most impact on biological activity. *MDLcompress* improves on our previous algorithm in execution time through an innovative data structure, and in specificity of motif detection (compression) through improved heuristics. An *MDLcompress* analysis of 144 over expressed genes from the breast cancer cell line BT474 has identified novel motifs, including potential microRNA (miRNA) binding sites that are candidates for experimental validation.

1. INTRODUCTION

The discovery of RNA interference (RNAi) [1] and certain of its endogenous mediators, the microRNAs (miRNAs), has catalyzed a revolution in biology and medicine [2, 3]. MiRNAs are transcribed as long (∼1000 nt) "pri-miRNAs," cut into small (∼70 nt) stem-loop "precursors," exported into the cytoplasm of cells, and processed into short (∼20 nt) single-stranded RNAs, which interact with multiple proteins to form a superstructure known as the RNA-induced silencing complex (RISC). The RISC binds to sequences in the 3′untranslated region (3′UTR) of mature messenger RNA (mRNA) that are partially complementary to the miRNA. Binding of the RISC to a target mRNA induces inhibition of protein translation by either (i) inducing cleavage of the mRNA or (ii) blocking translation of the mRNA. MiRNAs therefore represent a nonclassical mechanism for regulation of gene expression.

MiRNAs can be potent mediators of gene expression, and this fact has lead to large-scale searches for the full complement of miRNAs and the genes that they regulate. Although it is believed that all information about a miRNA's targets is encoded in its sequence, attempts to identify targets by informatics methods have met with limited success, and the requirements on a target site for a miRNA to regulate a cognate mRNA are not fully understood. To date, over 500 distinct miRNAs have been discovered in humans, and estimates of the total number of human miRNAs range well into the thousands. Complex algorithms to predict which specific genes these miRNAs regulate often yield dozens or hundreds of distinct potential targets for each miRNA [4–6]. Because of the technical difficulty of testing, all potential targets of a single miRNA, there are few, if any, miRNAs whose activities have been thoroughly characterized in mammalian cells. This problem is of singular importance because of evidence suggesting links between miRNA expression and human disease, for example chronic lymphocytic leukemia and lung cancer [7, 8]; however, the genes affected by these changes in miRNA expression remain unknown.

MiRNA genes themselves were opaque to standard informatics methods for decades in part because they are primarily localized to regions of the genome that do not

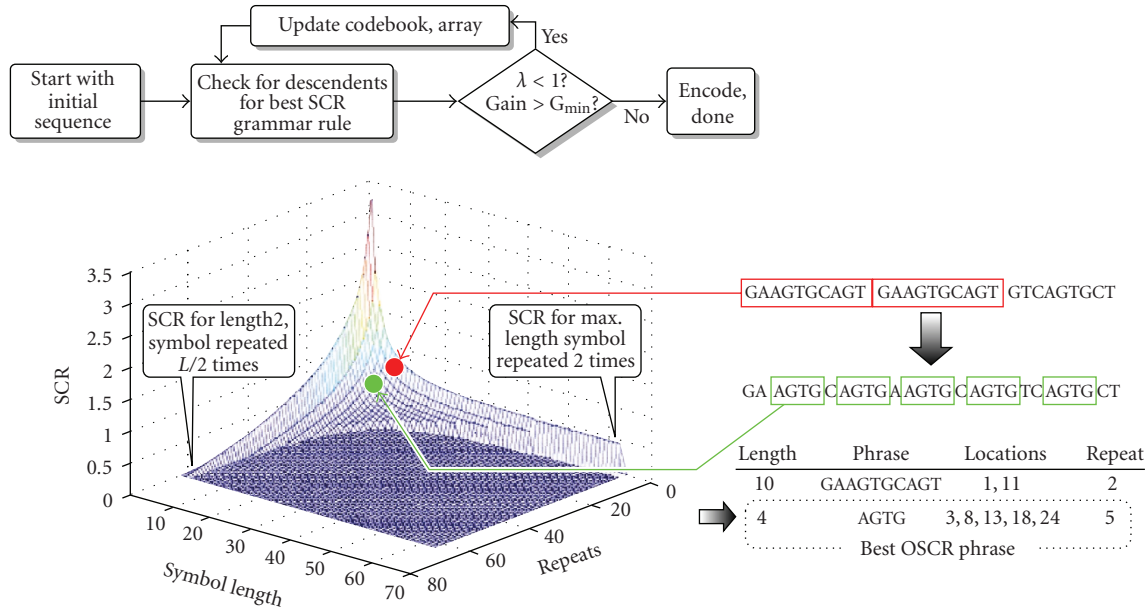


Figure 1: The OSCR algorithm. Phrases that recursively contribute most to sequence compression are added to the model first. The motif AGTG is the first selected and added to OSCR's MDL model. A longest match algorithm would not call out this motif.

code for protein. Informatics techniques designed to identify protein-coding sequences, transcription factors, or other known classes of sequence did not resolve the distinctive signatures of miRNA hairpin loops or their target sites in the 3′UTRs of protein-coding genes. In this sense, apart from comparative genomics, sequence analysis methods tend to be best at identifying classes of sequence whose biological significance is already known.

Minimum description length (MDL) principles [9] offer a general approach to de novo identification of biologically meaningful sequence information with a minimum of assumptions, biases, or prejudices. Their advantage is that they address explicitly the cost capability for data analysis without over fitting. The challenge of incorporating MDL into sequence analysis lies in (a) quantification of appropriate model costs and (b) tractable computation of model inference. A grammar inference algorithm that infers a two-part minimum description length code was introduced in [10], applied to the problem of information security in [11] and to miRNA target detection in [12]. This optimal symbol compression ratio (OSCR) algorithm produces "meaningful models" in an MDL sense while achieving a combination of model and data whose descriptive size together represents an estimate of the Kolmogorov complexity of the dataset [13]. We anticipate that this capacity for capturing the regularity of a data set within compact, meaningful models will have wide application to DNA sequence analysis.

MDL principles were successfully applied to segment DNA into coding, noncoding, and other regions in [14]. The normalized maximum likelihood model (an MDL algorithm) [15] was used to derive a regression that also achieves near state-of-the-art compression. Further MDL-related approaches include the "greedy offline"—GREEDY—algorithm [16] and DNA Sequitur [17, 18]. While these grammar-based codes do not achieve the compression of DNACompress [19] (see [20] for a comparison and additional approach using dynamic programming), the structure of these algorithms is attractive for identifying biologically meaningful phrases. The compression achieved by our algorithm exceeds that of DNA Sequitur while retaining a two-part code that highlights biologically significant phrases. Differences between MDLcompress and GREEDY will be discussed later. The deep recursion of our approach combined with its two-part coding makes our algorithm uniquely able to identify biologically meaningful sequence de novo with a minimal set of assumptions. In processing a gene transcript, we selectively identify sequences that are (i) short but occur frequently (e.g., codons, each 3 nucleotides) and (ii) sequences that are relatively long but occur only a small number of times (e.g., miRNA target sites, each ∼20 nucleotides or more). An example is shown in Figure 1, where given the input sequence shown, OSCR highlights the short motif AGTG that occurs five times, over a longer sequence that occurs only twice. Other model inference strategies would bypass by this short motif.

In this paper, we describe initial results of miRNA analysis using OSCR and introduce improvements to OSCR that reduce execution time and enhance its capacity to identify biologically meaningful sequence. These modifications, some of which were first introduced in [21], retain the deep recursion of the original algorithm but exploit novel data structures that make more efficient use of time and memory by gathering phrase statistics in a single pass and subsequently selecting multiple codebook phrases. Our data structure incorporates candidate phrase frequency information and pointers identifying location of candidate phrases in the sequence, enabling efficient computation. MDL model inference refinement is achieved by improving heuristics,

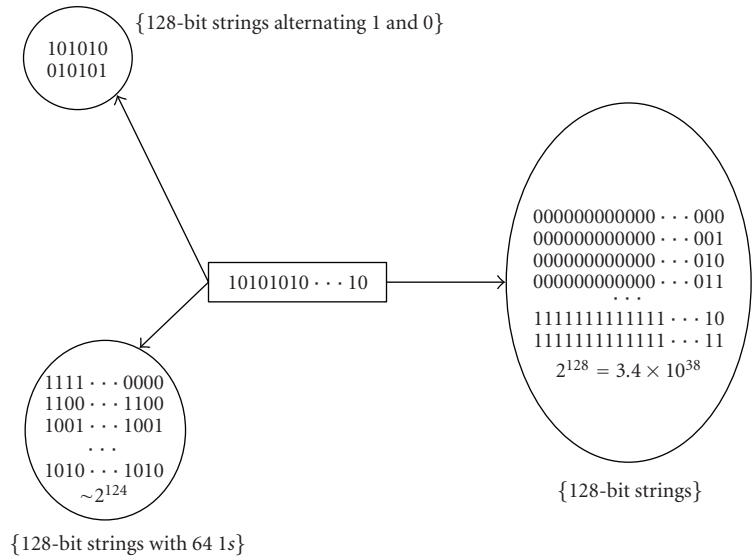


Figure 2: Two-part representations of a 128-bit string. As the length of the model increases, the size of the set including the target string decreases.

harnessing redundancies associated with palindrome data, and taking advantage of local sequence similarity. Since it now employs a suite of heuristics and MDL compression methods, including but not limited to the original symbol compression ratio (SCR) measure, we refer to this improved algorithm as MDLcompress, reflecting its ability to apply MDL principles to infer grammar models through multiple heuristics.

We hypothesized that MDL models could discover biologically meaningful phrases within genes, and after summarizing briefly our previous work with OSCR, we present here the outcome of an MDLcompress analysis of 144 genes overexpressed in the breast cancer cell line, BT474. Our algorithm has identified novel motifs including potential miRNA binding sites that are being considered for in vitro validation studies. We further introduce a "bits per nucleotide" MDL weighting from MDLcompress models and their inherent biologically meaningful phrases. Using this weighting, "susceptible" areas of sequence can be identified where an SNP disproportionately affects MDL cost, indicating an atypical and potentially pathological change in genomic information content.

2. MINIMUM DESCRIPTION LENGTH (MDL) PRINCIPLES AND KOLMOGOROV COMPLEXITY

MDL is deeply related to Kolmogorov complexity, a measure of descriptive complexity contained in an object. It refers to the minimum length l of a program such that a universal computer can generate a specific sequence [13]. Kolmogorov complexity can be described as follows, where φ represents a universal computer, p represents a program, and x represents a string:

$$K_\varphi(x) = \left\{ \min_{\varphi(p)=x} l(p) \right\}. \tag{1}$$

As discussed in [22], an MDL decomposition of a binary string x considering finite set models can be separated into two parts,

$$K_\varphi(x) \stackrel{+}{=} \{K(S) + \log_2 |S|\}, \tag{2}$$

where again $K_\varphi(x)$ is the Kolmogorov complexity for string x on universal computer φ. S represents a finite set of which x is a typical (equally likely) element. The minimum possible sum of descriptive cost for set S (the model cost encompassing all regularity in the string) and the log of the sets cardinality (the required cost to enumerate the equally likely set elements) correspond to an MDL two-part description for string x, a model portion that describes all redundancy in the string, and a data portion that uses the model to define the specific string. Figure 2 shows how these concepts are manifest in three two-part representations of the 128 binary string $101010 \cdots 10$. In this representation, the model is defined in English language text that defines a set, and the $\log_2$ of the number of elements in the defined set is the data portion of the description. One representation would be to identify this string by an index of all possible 128-bit strings. This involves a very small model description, but a data description of 128 bits, so no compression of descriptive cost is achieved. A second possibility is to use additional model description to restrict the set size to contain only strings with equal number of ones and zeros, which reduces the cardinality of the set by a few bits. A more promising approach will use still more model description to identify the set of alternating pattern of ones and zeros that could contain only two strings. Among all possible two-part descriptions of this string the combination that minimizes the two-part descriptive cost is the MDL description.

This example points out a major difference between Shannon entropy and Kolmogorov complexity. The first-order empirical entropy of the string $101010 \cdots 10$ is very

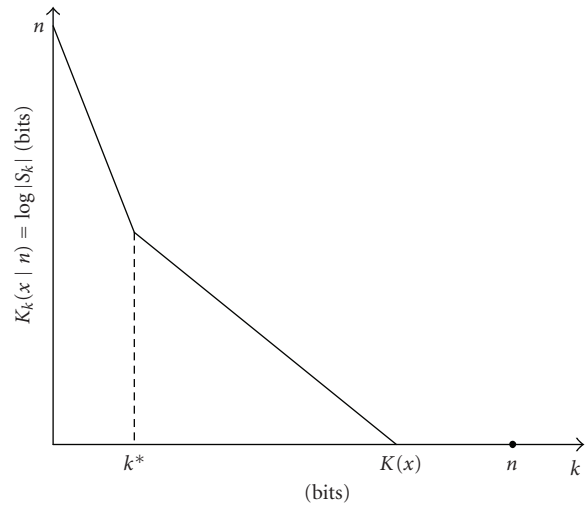


Figure 3: This figure shows the Kolmogorov structure function. As the model size (k) is allowed to increase, the size of the set (n) including string x with an equally likely probability decreases. k^* indicates the value of the Kolmogorov minimum sufficient statistic.

high, since the numbers of ones and zeros are equal. However, intuitively the regularity of the string makes it seem strange to call it random. By considering the model cost, as well as the data costs of a string, MDL theory provides a formal methodology that justifies objectively classifying a string as something other than a member of the set of all 128 bit binary. These concepts can be extended beyond the class of models that can be constructed using finite sets to all computable functions [22].

The size of the model (the number of bits allocated to spelling out the members of set S) is related to the Kolmogorov structure function, $\hbar$ (see [23]). $\hbar$ defines the smallest set, S, that can be described in at most k bits and contains a given string x of length n,

$$\hbar_k(x^n \mid n) = \min_{p:l(p)<k, U(p,n)=S} \{\log_2 |S|\}. \tag{3}$$

Cover [23] has interpreted this function as a minimum sufficient statistic, which has great significance from an MDL perspective. This concept is shown graphically in Figure 3. The cardinality of the set containing string x of length n starts out as equal to n when $k = 0$ bits is used to describe set S (restrict its size). As k increases, the cardinality of the set containing string x can be reduced until a critical value k^* is reached which is referred to as the Kolmogorov minimum sufficient statistic, or algorithmic minimum sufficient statistic [22]. At k^*, the size of the two-part description of string x equals $K_\varphi(x)$ within a constant. Increasing k beyond k^* will continue to make possible a two-part code of size $K_\varphi(x)$, eventually resulting in a description of a set containing the single element x. However, beyond k^*, the increase in the descriptive cost of the model, while reducing the cardinality of the set to which x belongs, does not decrease the string's overall descriptive cost.

The optimal symbol compression ratio (OSCR) algorithm is a grammar inference algorithm that infers a two-part minimum description length code and an estimate of the algorithmic minimum sufficient statistic [10, 11]. OSCR produces "meaningful models" in an MDL sense, while achieving a combination of model plus data whose descriptive size together estimate the Kolmogorov complexity of the data set. OSCR's capability for capturing the regularity of a data set into compact, meaningful models has wide application for sequence analysis. The deep recursion of our approach combined with its two-part coding nature makes our algorithm uniquely able to identify meaningful sequences without limiting assumptions.

The entropy of a distribution of symbols defines the average per symbol compression bound in bits per symbol for a prefix free code. Huffman coding and other strategies can produce an instantaneous code approaching the entropy in the limit of infinite message length when the distribution is known. In the absence of knowledge of the model, one way to proceed is to measure the empirical entropy of the string. However, empirical entropy is a function of the partition and depends on what substrings are grouped together to be considered symbols. Our goal is to optimize the partition (the number of symbols, their length, and distribution) of a string such that the compression bound for an instantaneous code, (the total number of encoded symbols R time entropy H_s) plus the codebook size is minimized. We define the approximate model descriptive cost M to be the sum of the lengths of unique symbols, and total descriptive cost D_p as follows:

$$M \equiv \sum_i l_i, \qquad D_p \equiv M + R \cdot H_s. \tag{4}$$

While not exact (symbol delimiting "comma costs" are ignored in the model, while possible redundancy advantages are not considered either), these definitions provide an approximate means of breaking out MDL costs on a per symbol basis. The analysis that follows can easily be adapted to other model cost assumptions.

2.1. Symbol compression ratio

In seeking to partition the string so as to minimize the total string descriptive length D_p, we consider the length that the presence of each symbol adds to the total descriptive length and the amount of coverage of total string length L that it provides. Since the probability of each symbol, p_i, is a function of the number of repetitions of each symbol, it can be easily shown that the empirical entropy for this distribution reduces to

$$H_s = \log_2(R) - \frac{1}{R} \sum_i r_i \log_2 (r_i). \tag{5}$$

Thus, we have

$$\begin{aligned} D_p &= R \log_2(R) + \sum_i l_i - r_i \log_2 (r_i), \quad \text{with} \\ R \log_2(R) &= \sum_i r_i \log_2(R) = \log_2(\hat{R}) \sum_i r_i, \end{aligned} \tag{6}$$

where $\log_2(\hat{R})$ is a constant for a given partition of symbols. Computing this estimate based on the partition in hand

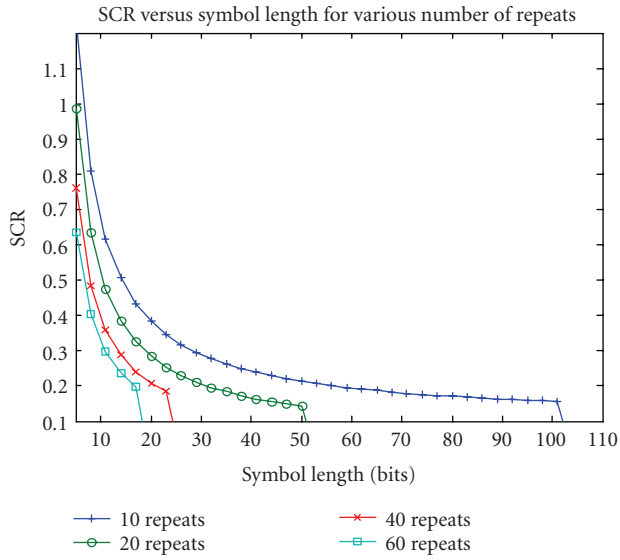


FIGURE 4: SCR versus symbol length for 1024-bit string.

enables a per-symbol formulation for D_p and results in a conservative approximation for $R\log_2(R)$ over the likely range of R. The per-symbol descriptive cost can now be formulated:

$$d_i = r_i[\log_2(\hat{R}) - \log_2(r_i)] + l_i. \tag{7}$$

Thus, we have a heuristic that conservatively estimates the descriptive cost of any possible symbol in a string considering both model and data (entropy) costs. A measure of the compression ratio for a particular symbol is simply the descriptive length of the string divided by the length of the string "covered" by this symbol. We define the symbol compression ratio (SCR) as

$$\lambda_i = \frac{d_i}{L_i} = \frac{r_i[\log_2(\hat{R}) - \log_2(r_i)] + l_i}{l_i r_i}. \tag{8}$$

This heuristic describes the "compression work" a candidate symbol will perform in a possible partition of a string. Examining SCR in Figure 4, it is clear that good symbol compression ratio arises in general when symbols are long and repeated often. But clearly, selection of some symbols as part of the partition is preferred to others. Figure 4 shows how symbol compression ratio varies with the length of symbols and number of repetitions for a 1024 bit string.

3. OSCR ALGORITHM

The optimal symbol compression ratio (OSCR) algorithm forms a partition of string S into symbols that have the best symbol compression ratio (SCR) among possible symbols contained in S. The algorithm is as follows.

(1) Starting with an initial alphabet, form a list of substrings contained in S, possibly with user-defined constraints on minimum frequency and/or maximum length, and note the frequency of each substring.

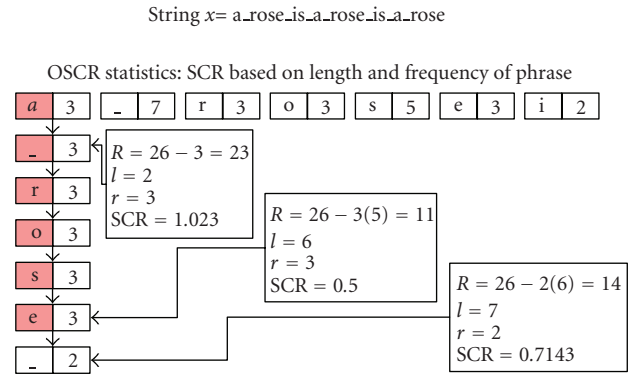


FIGURE 5: OSCR example.

(2) Calculate the SCR for all substrings. Select the substring from this set with the smallest SCR and add it to the model M.
(3) Replace all occurrences of the newly added substring with a unique character.
(4) Repeat steps 1 through 3 until no suitable substrings are found.
(5) When a full partition has been constructed, use Huffman coding or another coding strategy to encode the distribution, p, of symbols.

The following comments apply.

(1) This algorithm progressively adds symbols that do the most compression "work" among all the candidates to the code space. Replacement of these symbols leftmost-first will alter the frequency of remaining symbols.
(2) A less exhaustive search for the optimal SCR candidate is possible by concentrating on the tree branches that dominate the string or searching only certain phrase sizes.
(3) The initial alphabet of terminals is user supplied.

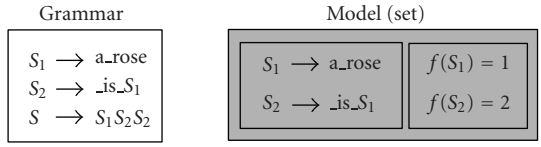
3.1. Example

Consider the phrase "a rose is a rose is a rose" with ASCII characters as the initial alphabet. The initial tree statistics and λ calculations provide the metrics shown in Figure 5. The numbers across the top indicate the frequency of each symbol, while the numbers along the left indicate the frequency of phrases.

Here we see that the initial string consists of seven terminals $\{a, \_, r, o, s, e, i\}$. Expanding the tree with substrings beginning with the terminal a shows that there are 3 occurrences of substrings:

$$\{a, a\_, a\_r, a\_ro, a\_ros, a\_rose\}, \tag{9}$$

but only 2 occurrences of longer substrings, for each of which λ values consequently increase, leaving the phrase $\{a\_rose\}$ the candidate with the smallest λ. Here we see the unique nature of the λ heuristic, which does not choose necessarily



Equally likely musings:

$$\text{TypicalSet} = \begin{Bmatrix} S_1S_2S_2 \\ S_2S_1S_2 \\ S_2S_2S_1 \end{Bmatrix} = \begin{Bmatrix} a\_\text{rose\_is\_}a\_\text{rose\_is\_}a\_\text{rose} \\ \text{\_is\_}a\_\text{rose\_}a\_\text{rose\_is\_}a\_\text{rose} \\ \text{\_is\_}a\_\text{rose\_is\_}a\_\text{rose\_}a\_\text{rose} \end{Bmatrix}$$

Figure 6: OSCR grammar example model summary.

the most frequently repeating symbol, or the longest match but rather a combination of length and redundancy. A second iteration of the algorithm produces the model described in Figure 6. Our grammar rules enable the construction of a typical set of strings where each phrase has frequency shown the model block of Figure 6. One can think of MDL principles applied in this way as analogous to the problem of finding an optimal compression code for a given dataset x with the added constraint that the descriptive cost of the codebook must also be considered. Thus, the cost of sending "priors" (a codebook or other modeling information) is considered in the total descriptive cost in addition to the descriptive cost of the final compressed data given the model.

The challenge of incorporating MDL in sequence analysis lies in the quantification of appropriate model costs and tractable computation of model inference. Hence, OSCR has been improved and optimized through additional heuristics and a streamlined architecture and renamed MDLcompress, which will be described in detail in later sections. MDLcompress forms an estimate of the strings algorithmic minimum sufficient statistic by adding bits to the model until no additional compression can be realized. MDLcompress retains the deep recursion of the original algorithm but improve speed and memory use through novel data structures that allow gathering of phrase statistics in a single pass and subsequent selection of multiple codebook phrases with minimal computation.

MDLcompress and OSCR are not alone in the grammar inference domain. GREEDY, developed by Apostolico and Lonardi [16], is similar to MDLcompress and OSCR, but differ in three major areas.

(1) MDLcompress is deeply recursive in that the algorithm does not remove phrases from consideration for compression after they have been added to the model. The "loss of compressibility" inherent in adding a phrase to the model was one of the motivations of developing the SCR heuristic—preventing a "too greedy" absorption of phases from preventing optimal total compression. With MDLcompress, since we look in the model as well for phrases to compress, we find that generally the total compression heuristic at each phase gives the best performance as will be discussed later.

(2) MDLcompress was designed with the express intent of estimating the algorithmic minimum sufficient statistic, and thus has more stringent separation of model and data costs and more specific model cost calculations resulting in greater specificity.

(3) As described in [21] and will be discussed in later sections, the computational architecture of MDLcompress differs from the suffix tree with counts architecture of GREEDY. Specifically, MDLcompress gathers statistics in a single pass and then updates the data structure and statistics after selecting each phrase as opposed GREEDY's practice of reforming the suffix tree with counts data structure at each iteration.

Another comparable grammar-based code is Sequitur, a linear time grammar inference algorithm [17, 18]. In this paper, we show MDLcompress to exceed Sequitur's ability to compress. However, it does not match Sequitur's linear run time performance.

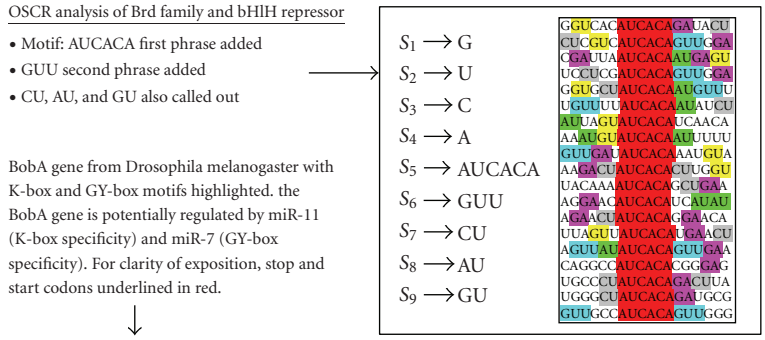
4. MIRNA TARGET DETECTION USING OSCR

In [12], we described our initial application of the OSCR algorithm to the identification of miRNA target sites. We selected a family of genes from Drosophila (fruit fly) that contain in their 3′UTRs conserved sequence structures previously described by Lai [24]. These authors observed that a highly-conserved 8-nucleotide sequence motif, known as a K-box (sense = 5′ cUGUGAUa 3′; antisense = 5′ uAUCACAg) and located in the 3′UTRs of Brd and bHLH gene families, exhibited strong complementarity to several fly miRNAs, among them miR-11. These motifs exhibited a role in posttranscriptional regulation that was at the time unexplained.

The OSCR algorithm constructed a phrasebook consisting of nine motifs, listed in Figure 7 (top) to optimally partition the adjacent set of sequences, in which the motifs are color coded. The OSCR algorithm correctly identified the most redundant antisense sequence (AUCACA) from the several examples it was presented.

The input data for this analysis consists of 19 sequences, each 18 nucleotides in length (Figure 7). From these sequences, OSCR generated a model consisting of grammar "variables" S_1 through S_4 that map to individual nucleotides (grammar "terminals"), the variable S_5 that maps to the nucleotide sequence, AUCACA, and four shorter motifs S_6–S_9. The phrase S_5 turns out to be a putative target of several different miRNAs, including miR-2a, miR-2b, miR-6, miR-13a, miR-13b, and miR-11. OSCR identified as S_9 a 2 nucleotide sequence (5′ GU 3′) that is located immediately downstream of the K-box motif. The new consensus sequence would read 5′ AUCACAGU 3′ and has a greater degree of homology to miR-6 and miR-11 than to other *D. melanogaster* miRNAs. In vivo studies performed subsequent to the original Lai paper demonstrated the specificity of miR-11 activity on the Bob-A,B,C, E(spl)ma, E(spl)m4, and E(spl)md genes [25].

In a separate analysis, we applied OSCR to the sequence of an individual fruit fly gene transcript, BobA (accession NM 080348; Figure 7, bottom). Only the BobA transcript



1 aacaguucuccauccgagcagaucauaaguaaccaaccugcaaa**aug**uucaccgaaaccg
61 cucuuguuuccaacuucaauggagugacagagaagaaaaucucuuaccggcgccuccacca
121 accugaagaagcugcugaagaccaucaagaaggucuucaagaacuccaagccuucgaagg
181 agauuccgaucccccaacaucaucuacucuugcaauacugaggaggagcaccagaauuggc
241 ucaacgaacaacuggaggccauggcaauccaucuucac**uga**guucuucugggacaucccc
301 cuccaucgaguau**cugugaug**ugacccgaucaaaaggucuauaaaucggcacuccggcuu
361 uaauauccaa**cugugaug**acgagaacacaagacugacugacuugugugccuuggagguga
421 caaaguucgucgccucugccaacuguacauaucaaacuagcugcuaaaaugucuucaauu
481 augcuuuaauguagucuaaguuaguauuuaucau**ugucuuccau**uaguuuaagaaaaucau
541 **ugucuuccau**guuuguuuguuaggguaaaaaaaacuagcuuaagaauaaaaaucccucgc
601 ggaaagaaaacaau

Figure 7: Motif analysis of 19 sequences each of which is believed to contain a single target site for miR-11 from fruit fly. (Top) OSCR adds the variable S_5 to its MDL codebook, the K-box motif, which has been shown to be a miRNA target site for miR-11. (Bottom) Full sequence of BobA gene transcript with K-box and GY box motifs underlined in blue text. The K-box motif (CUGUGAUG) is a target site for miR-11 and the GY-box motif (UGUCUUCCAU) is a target site for miR-7.

itself entered this second analysis, which was performed independently of the multisequence analysis described in the paragraph above. The sense sequence of BobA is displayed in Figure 2 with the 5′UTR indicated in green; the 237 nucleotides (79 codons) of the coding sequence in red; and the 3′UTR in blue. OSCR identified the underlined motifs, (cugugaug) and (ugucuuccau). These two motifs turn out not only to be conserved among multiple Drosophila subspecies, but also to be targets of two distinct miRNAs: the K-box motif (cugugaug) is a target of miR-11 and the GY-box (ugucuuccau) a target of miR-7. Although we did not perform OSCR analysis on any additional genes, this motif had been identified previously in several 3′UTRs, including those of BobA, E(spl)m3, E(spl)m4, E(spl)m5, and Tom [23, 24]. The BobA gene is particularly sensitive to miR-7. Mutants of the BobA gene with base-pair disrupting substitutions at both sites of interaction with miR-7 yielded nearly complete loss of miR-7 activity [25] both in vivo and in vitro. These observations are consistent with studies from [26, 27] that reveal specific sequence-matching requirements for effective miRNA activity in vitro.

In summary, the OSCR algorithm identified (i) a previously-known 8-nucleotide sequence motif in 19 different sequence and (ii) in an entirely independent analysis, identified 2 sequence motifs, the K-box and GY-box, within the BobA gene transcript. We now describe innovative refinements to our MDL-based DNA compression algorithm with the goal of improved identification and analysis of biologically meaningful sequence—particularly miRNA targets related to breast cancer.

5. MDLcompress

The new MDLcompress algorithmic tool retains the fundamental element of OSCR—deeply—recursive heuristic-based grammar inference, while trading computational complexity for space complexity to decrease execution time. The compression and hence the ability of the algorithm to identify specific motifs (which we hypothesize to be of potential biological significance) have been enhanced by new heuristics and an architecture that searches not only the sequence but also the model for candidate phrases. The performance has been improved by gathering statistics about potential code words in a single pass and forming and maintaining simple matrix structures to simplify heuristic calculations. Additional gains in compression are achieved by tuning the algorithm to take advantage of sequence-specific features such as palindromes, regions of local similarity, and SNPs.

5.1. Improved SCR heuristic

MDLcompress uses steepest-descent stochastic-gradient methods to infer grammar-based models based upon phrases that maximize compression. It estimates an algorithmic minimum sufficient statistic via a highly recursive algorithm that identifies those motifs enabling maximal compression. A critical innovation in the OSCR algorithm was the use of a heuristic, the symbol compression ratio (SCR), to select phrases. A measure of the compression ratio for a particular symbol is simply the descriptive length of the string divided by number of symbols—grammar variables and terminals

encoded by this symbol in the phrasebook. We previously defined the SCR for a candidate phrase i as

$$\lambda_i = \frac{d_i}{L_i} = \frac{r_i[\log_2(R) - \log_2(r_i)] + l_i}{l_i r_i} \tag{10}$$

for a phrase of length l_i, repeated r_i times in a string of total length L, with R denoting the total number of symbols in the candidate partition. The numerator in the equation above consists of the MDL descriptive cost of the phrase if added to the model and encoded, while the denominator consists of an estimate of the unencoded descriptive cost of the candidate phrase. This heuristic encapsulates the net gain in compression per symbol that a candidate phrase would contribute if it were to be added to the model.

While (10) represents a general heuristic for determining the partition of a sequence that provides the best compression, important effects are not taken into account by this measure. For example, adding new symbols to a partition increases the coding costs of other symbols by a small amount. Furthermore, for any given length and frequency, certain symbols ought to be preferred over others, because of probability distribution effects. Thus, we desire an SCR heuristic that more accurately estimates the potential symbol compression of any candidate phrases.

To this end, we can separate the costs accounted for in (10) into three parameters: (i) entropy costs (costs to represent the new phrase in the encoded string); (ii) model costs (costs to add the new phrase to the model); and (iii) previous costs (costs to represent the substring in the string previously). The SCR of [10, 11, 28] breaks these costs down as follows:

$$C_h = R_i \cdot \log\left(\frac{\hat{R}}{R_i}\right), \tag{11}$$

$$\begin{aligned} C_m &= l_i, \\ C_p &= l_i R_i, \end{aligned} \tag{12}$$

where $\hat{R}$ is the length of the string after substitution, l_i is the length of the code phrase, L is the length of the model, and R_i is the frequency of the code phrase in the string. An improved version of this heuristic, SCR_2006, provides a more accurate description of the compression work by eliminating some of the simplifying assumptions made earlier. Entropy costs (11) remain unchanged. However, increased accuracy can be achieved by more specific costs for the model and previous costs. For previous costs we consider the sum of the costs of the substrings that comprise the candidate phrase

$$C_p = R_i \cdot \sum_{j=1}^{l_i} \log\left(\frac{\hat{R}'}{r_j}\right), \tag{13}$$

where $\hat{R}'$ is the total number of symbols without the formation of the candidate phrase and r_j is the frequency of the jth symbol in the candidate phrase. Model costs require a method for not only spelling out the candidate phrase but also the cost of encoding the length of the phrase to be described. We estimate this cost as

$$C_m = M(l_i) + \sum_{j=1}^{l_i} \log\left(\frac{\hat{R}'}{r_j}\right), \tag{14}$$

where $M(L)$ is the shortest prefix encoding for the length phrase. In this way we achieve both a practical method for spelling out the model for implementation and an online method for determining model costs that relies only on known information. Since new symbols will add to the cost of other symbols simply by increasing the number of symbols in the alphabet, we specify an additional cost that reflects the change in costs of substrings that are not covered by candidate phrase. The effect is estimated by

$$C_o = (\hat{R} - R_i) \cdot \log\left(\frac{L+2}{L+1}\right). \tag{15}$$

This provides a new, more accurate heuristic as follows:

$$\text{SCR\_2006} = \frac{C_m + C_h + C_o}{C_p}. \tag{16}$$

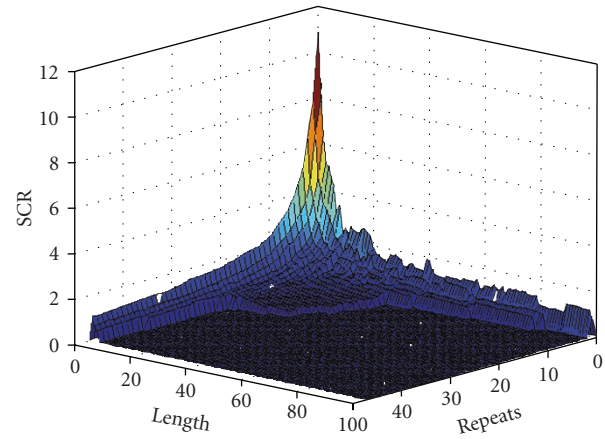


Figure 8: Symbol compression ratio (vertical axis) as a function of phrase length and number of occurrences (horizontal axes) for the first phrase encountered of a given length and frequency. The variation indicates our improved heuristic is providing benefit by considering descriptive cost of specific phrases based on the grammars and terminals contained in the phrase, not just length and number of occurrences.

Figure 8 shows a plot of SCR_2006 versus length and number of repeats for a specific sequence, where the first phrase of a given length and number of repeats is selected. Notice that the lowest SCR phrase is primarily a function of number of repeats and length, but also includes some variation due to other effects. Thus, we have improved the SCR heuristic to yield a better choice of phrase to add at each iteration.

5.2. Additional heuristics

In addition to SCR, two alternative heuristics are evaluated to determine the best phrase for MDL learning: longest match

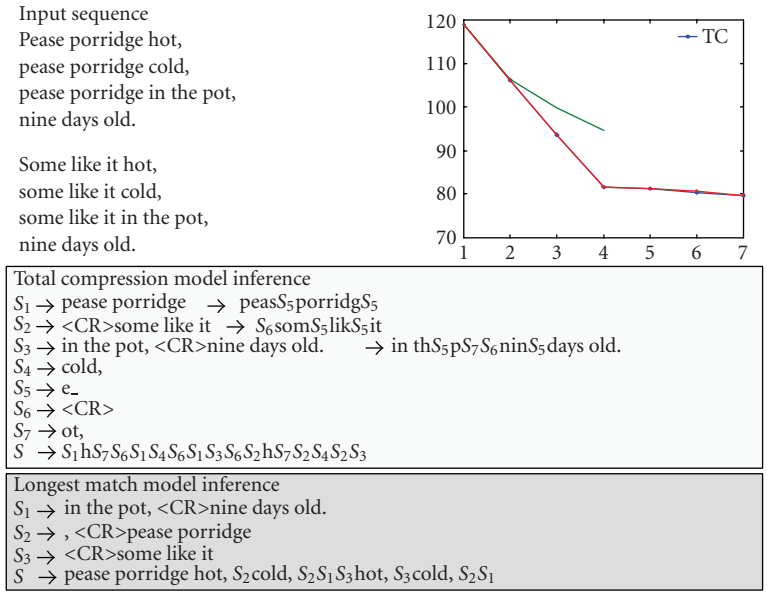


Figure 9: MDLcompress model-inferred grammar for the input sequence "pease porridge" using total compression (TC) and the longest match (LM) heuristics. Both the SCR and TC heuristics achieve the same total compression and both exceed the performance of LM. Subsequent iterations enable MDLcompress to identify phrases, yielding further compression of the TC grammar model.

(LM) and total compression (TC). Both of these heuristics leverage the gains described above by considering the entropy of specific variables and terminals when selecting candidate phrases. In LM, the longest phrase is selected for substitution, even if only repeated once. This heuristic can be useful when it is anticipated that the importance of a codeword is proportional to its length. MDLcompress can apply LM to greater advantage than other compression techniques because of its deep recursion—when a long phrase is added to codebook, its subphrases, rather than being disqualified, remain potential candidates for subsequent phrases. For example, if the longest phrase merely repeats the second longest phrase three times, MDLcompress will nevertheless identify both phrases.

In TC, the phrase that leads to maximum compression at the current iteration is chosen. This "greedy" process does not necessarily increase the SCR, and may lead to the elimination of smaller phrases from the codebook. MDLcompress, as explained above, helps temper this misbehavior by including the model in the search space of future iterations. Because of this "deep recursion" phrases in both the model and data portions of the sequence are considered as candidate codewords at each iteration-MDLcompress yields improved performance over the GREEDY algorithm [16]. As with all MDL criteria, the best heuristics for a given sequence is the approach that best compresses the data. The TC gain is the improvement in compression achieved by selecting a candidate phrase and can be derived from the SCR heuristic by removing the normalization factor. Examples of MDLcompress operating under different heuristics or combinations of heuristics are shown in Figures 9 and 10. Under our improved architecture, the best compression seems to usually be achieved in TC mode, which we attribute to the fact

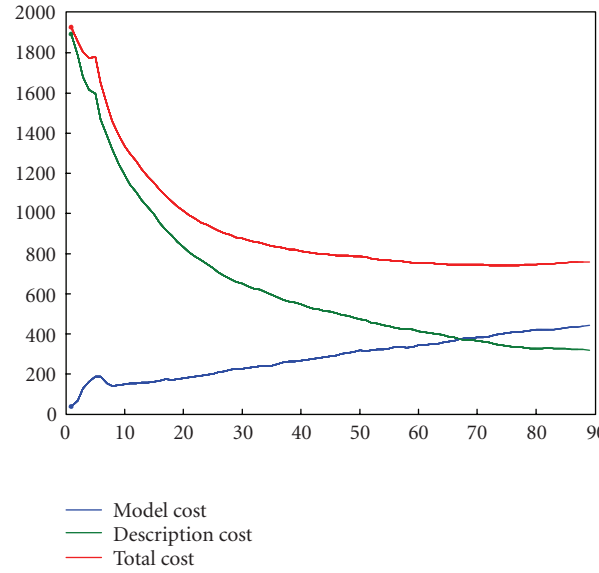


Figure 10: The compression characteristic of MDLcompress using the hybrid heuristics longest match, followed by total compress after the longest match heuristic ceases to provide compression.

that we search the model as well as remaining sequence for candidate phrases, reducing the need for and benefit from the SCR heuristic. By comparison, SEQUITUR [17] forms a grammar of 13 rules consisting of 74 symbols. Thus, using MDLcompress TC we achieve better compression with a grammar model of approximately half the size.

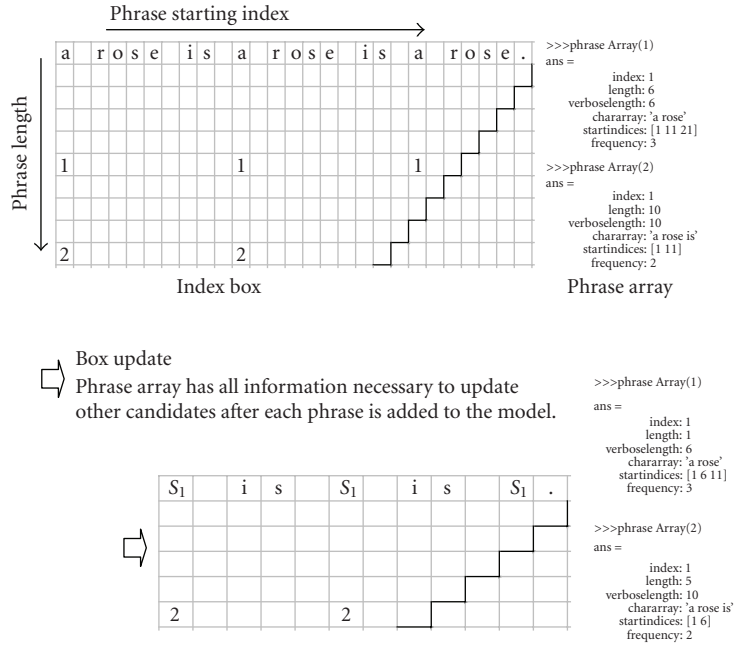


Figure 11: The data structures used in MDLcompress allow constant time selection and replacement of candidate phrases. In the top of the figure is the initial index matrix and phrase array. After adding "a rose" for the model, MDLcompress can generate the new index box and phrase array, shown in the bottom half, in constant time.

5.3. Data structures

A second improvement of MDLcompress over OSCR is the improvement to execution time to allow analysis of much longer input strings, such as DNA sequences. This is achieved through trading off memory usage and runtime by using matrix data structures to store enough information about each candidate phrase to calculate the heuristic and update the data structures of all remaining candidate phrases. This allows us to maintain the fundamental advantage of OSCR and algorithms such as GREEDY [16] that compression is performed based upon the global structure of the sequence, rather than by the phrases that happen to be processed first, as in schemes such as Sequitur, DNA Sequitur, and Lempel-Ziv. We also maintain an advantage over the GREEDY algorithm by including phrases added to our MDL model and the model space itself in our recursive search space.

During the initial pass of the input, MDLcompress generates an $l_{\max}$ by L matrix, where entry $M_{i,j}$ represents the substring of length i beginning at index j. This is a sparse matrix with entries only at locations that represent candidates for the model. Thus, substrings with no repeats and substrings that only ever appear as part of a longer substring are represented with a 0. Matrix locations with positive entries represent the index into an array with many more details for that specific substring. In the example in Figure 11, "a rose" appears three times in the input. In each location of the matrix corresponding to this substring is a 1, and the first element in the phrase array has the length, frequency, and starting index for all occurrences of the substring. A similar element exists for "a rose is" but not exist for "a rose" since that only appears as a substring of the first candidate.

During the phrase selection part of each iteration, MDLcompress only has to search through phrase array, calculating the heuristic for each entry. Once a phrase is selected, the matrix is used to identify overlapping phrases, which will have their frequency reduced by the substitution of a new symbol for the selected substring. While there may be many phrases in the array that are updated, only local sections of the matrix are altered, so overall only a small percentage of the data structure is updated. This technique is what allows MDLcompress to execute efficiently even with long input sequences, such as DNA.

5.4. Performance bounds

The execution of MDLcompress is divided into two parts: the single pass to gather statistics about each phrase and the subsequent iterations of phrase selection and replacement. Since simple matrix operations are used to perform phrase selection and replacement, the first pass of statistics gathering almost entirely dominates both the memory requirements and runtime.

For strings with input length, L, and maximum phrase length, $l_{\max}$, the memory requirements of the first pass are bounded by the product $L * l_{\max}$ and subsequent passes require less memory as phrases are replaced by (new) individual symbols. Since the user can define a constraint on $l_{\max}$, memory use can be restricted to as little as $O(L)$, and will never exceed $O(L^2)$. On platforms with limited memory where long phrases are expected to exist, the LM heuristic can be used in a simple preprocessing pass to identify and replace any phrases longer than the system can handle in the standard matrix described above. Because MDLcompress

Table 1

Genes	DNACompress (bits/nucleotide)	Sequitur	DNASequitur	*MDLcompresss*
HUMDYSTROP	1.91	2.34	2.2	1.95
HUMGHCSA	1.03	1.86	1.74	1.49
HUMHBB	1.79	2.20	2.05	1.92
HUMHDABCD	1.80	2.26	2.12	1.92
HUMPRTB	1.82	2.22	2.14	1.92
CHNTXX	1.61	2.24	2.12	1.95

inspects the model when searching for subsequent phrases, this technique has minimal negative effect on overall compression.

The runtime of the first pass depends directly on L, $l_{\max}$, average phrase length l_{avg}, and average number of repeats of selected phrases, r_{avg}. The unclear relationship between $l_{\max}$, l_{avg}, r_{avg}, and L makes deriving guaranteed performance bounds difficult. As a simple upper bound, we can note that the product $l_{\text{avg}} * r_{\text{avg}}$ must be less than L, and the maximum phrase length must be less than $L/2$, yielding a performance bound of $O(L^3)$. In practice, a memory constraint limits $l_{\max}$ to a constant independent of L, and $l_{\text{avg}} * r_{\text{avg}}$ was approximately constant and much smaller than L. Thus, the practical performance bound was $O(L)$.

The runtime of the second part of the algorithm, selection and replacement of compressible phrases, is simply the sum of the time to identify the best phrase and to update the matrices for the next iteration, multiplied by the number of iterations. An upper bound on these is $O(L^2)$, but again practical performance is much better. In this DNA application where 144 genes were analyzed, the number of candidate phrases, the average number of affected phrases, and the number of iterations all were independent of input length, and the selection and replacement phase ran in constant time.

5.5. Enhancements for DNA compression

When a symbol sequence is already known to be DNA, several "priors" can be incorporated into the model inference algorithm that may lead to improved compression performance. These assumptions relate to types of structure that are typical of naturally occurring DNA sequence. By tuning our algorithm to efficiently code for these mechanisms, we are essentially incorporating these priors into our model inference algorithm "by hand." We consider these assumptions to be small and within the "big O" constant inherent in translating between universal computers.

6. REVERSE-COMPLEMENT MATCHES

As in DNA Sequitur, the search for and grammar encoding of reverse-complement matches is readily implemented by adding the reverse-complement of a phrase to the MDLcompress model and taking account of the frequency of the phrase and its reverse-complement in motif selection.

7. POST PROCESSING

After the MDLcompress model has been created, two methods possibilities for further compression are the following.

(1) Regions of Local similarity: it is sometimes most efficient to define a phrase as a concatenation of multiple shorter and adjacent phrases already in the codebook.
(2) Single nucleotide polymorphisms (SNPs): it is sometime most efficient to define a phrase as a single nucleotide alteration to another phrase already in the codebook.

8. COMPARISON TO OTHER GRAMMAR-BASED CODES

We compare MDLcompress with the state of the art in grammar-based compression: DNA Sequitur [18]. DNA Sequitur improves the Sequitur algorithm by enabling it to harness advantages of palindromes and by considering other grammar-based encoding techniques as discussed in [20]. Results are summarized in Table 1.

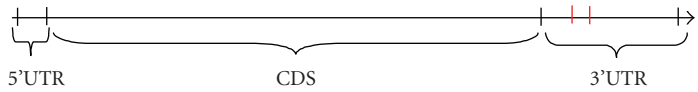
While compression is ultimately the best measure of algorithm's capacity to approximate Kolmogorov complexity, an additional feature of grammar-based codes is their two-part encoding, which separates the meaningful model from the data elements—an advantage we will discuss in more detail later. The results above make use of the total compression heuristic and harness the advantage of considering palindromes. Although we exceeded the compression of DNA Sequitur, DNACompress still achieves better compression; however it does not yield the two-part grammar code that identifies biologically significant phrases, which we will discuss next in the context of breast-cancer-related genes.

9. IDENTIFICATION OF MIRNA TARGETS USING MDLCOMPRESS

As shown in Figure 7, MDL algorithms can be used to identify miRNA target sites. We have also tested MDLcompress for the ability to identify miRNA target sites in known disease-related genes. The general approach is to analyze mRNA transcripts to identify short sequences that are

MDLcompress & LATS2: sequence elements in long 3'UTR

LOCUS NM_014572
Definition homo sapiens LATS, large tumor suppressor, homolog 2 (Drosophila) (LATS2), mRNA.



MDLcompress (of 3'UTR) output sequences

Sequence	Position in 3'UTR
1) aaaaaaaaaaaa	433, 445
2) agcacttatt	262, 362
3) aaacaggac	155, 172

Figure 12: Validation of MDLcompress performance. MDL compress identifies miRNA-372 and 373 target motif (AGCACTTATT) in LATS2 tumor suppressor gene as second phrase.

repeated and localized to the 3′UTR. Comparative genomics can be applied to increase our confidence that MDL phrases in fact represent candidate miRNA target sites, even if there are no known cognate miRNAs that will bind to that site.

As a test, we sought to determine if MDLcompress would have identified the miRNA binding site in the 3′UTR of the tumor suppressor gene, LATS2. A recent study, which used a function-based approach to miRNA target site identification, determined that LATS2 is regulated by miRNAs 372 and 373 [29]. Increased expression of these miRNAs led to down regulation of LATS2 and to tumorigenesis. The miRNA 372 and 373 target sequence (AGCACTTATT) is located in the 3′UTR of LATS2 mRNA and is repeated twice but was not identified with computation-based miRNA target identification techniques. Using the 3′UTR of LATS2 mRNA as an input, three code words were added to the MDLcompress model, using longest match mode as shown in Figure 12, the polyA tail, the miRNA 372 and 373 target sequence (AGCACTTATT), and a third phrase (AAACAGGAC) which we do not identify with any particular biological function at this time. This shows that analyzing genes of interest a priori with MDLcompress can produce highly relevant sequence motifs.

Since miRNAs regulate genes important for tumorigenesis and MDLcompress is able to identify these targets, it follows that MDLcompress could be used to directly identify genes that are important for tumorigenesis. To test this, we used a target rich set of 144 genes known to have increased expression patterns in ErbB2-positive breast cancer [30, 31] and compressed each gene mRNA sequence with MDLcompress running in longest match mode. A total of 93 phrases were added to MDLcompress codebooks resulting in compression of these genes. Of these phrases, 25 were found exclusively in the 3′UTRs of these genes. Since miRNAs interact more frequently with the 3′UTRs of mRNAs [32], we focused our analysis on these phrases, shown in Table 2.

The 25 3′UTR phrases were run through BLAST [33] searches of a database of 3′UTRs [34, 35] to determine level of conservation in human and other genomes. The phrases were also run against the miRBase database [36] using SSEARCH [37] to detect possible sequence similarities to known miRNAs. Finally, genes containing these phrases were targeted with shRNA constructs in an ErbB2-positive breast cancer cell line (BT474), as well as in normal mammary epithelial cells (HMEC), in order to identify their potential role in breast tumorigenicity. One MDLcompress phrase, AGAUCAAGAUC, found in the 3′UTR of the splicing factor arginine/serine-rich 7 (SFRS7) gene (a) was highly conserved, (b) resulted in miRBase matches to a small number of miRNAs that fulfill the minimum requirements of putative miRNA targets [32] (Figures 13(a) and 13(b)) in vitro data implicate this gene in breast cancer progression. More specifically, down regulation of SFRS7 by shRNAs in BT474 cells yielded a significant decrease in the proliferation marker alamarBlue (Biosource), but not in normal mammary epithelial cells (HMEC) (Figure 13(b)). In this experiment, cells were transiently transfected with miRNA-based-structure shRNA constructs [38] targeting the coding sequence of SFRS7, by using a lipid-based reagent (FuGENE 6, Roche). A plasmid construct expressing green fluorescent protein (MSCV-GFP) was cotransfected to the cells to normalize transfection efficiency [3]. shRNAs against the firefly luciferase gene was used as negative control. Although regulation by the specific miRNAs identified in our bioinformatics analysis still requires validation, these results suggest the possible differential regulation of this gene in breast cancer by a miRNA and that this gene is significant in cell proliferation, underscoring the potential for OSCR to identify sequence of biological interest.

10. ANALYSIS OF SINGLE NUCLEOTIDE POLYMORPHISMS

By definition, mutation of an essential nucleotide within a given miRNA's target sequence within an mRNA is expected to have a strong effect on the activity of the given miRNA on the target. If a nucleotide that is required for interaction of a miRNA with the mRNA is altered, the miRNA may cease to regulate that target, thereby enhancing expression of the mRNA and the protein it encodes. Alternatively, a

Table 2: 3′UTR MDLcompress phrases from 144 ErbB2-positive-related gene mRNA sequence.

Accession number	Number of repeats	Length	Phrase	Locations
NM_000442	2	13	tttctcttttcct	2835, 3091
NM_004265	2	10	tcagggaggg	2274, 2667
NM_004265	2	10	ccccccagct	2954, 3021
NM_004265	2	10	gcagaggcag	2255, 3051
NM_005324	2	12	ttttatttataa	1292, 1802
NM_005324	2	10	cagtttcctt	997, 1991
NM_005324	2	9	tttataata	627, 1055
NM_005930	2	11	tatttcaattt	2903, 2932
NM_005930	2	11	tatttttgctc	2733, 3809
NM_005930	2	10	gacaaatgtg	3064, 3250
NM_005930	2	10	ctttttttc	3425, 3689
NM_005930	2	10	ttggaacact	3750, 3787
NM_006148	2	13	gtgtgtgagtgtg	1951, 3654
NM_006148	2	12	ccccagtctcca	647, 1651
NM_006148	2	11	acttcttggtt	1067, 1290
NM_006148	2	11	cctcctgccca	1186, 1503
NM_006148	2	11	ccccatctctg	2147, 2302
NM_006148	2	11	ggaagcacagc	1545, 2447
NM_006148	2	11	tgtgggtgggg	2014, 2776
NM_006148	2	11	cctttctggcc	2812, 3759
NM_006148	2	10	ctccctcctc	1035, 1408
NM_006148	2	10	cagctaccgg	525, 1591
NM_006148	2	10	tcccctcccc	1464, 1828
NM_006148	2	10	gtggaggaag	2159, 2267
NM_006276	2	11	agatcaagatc	1010, 1091

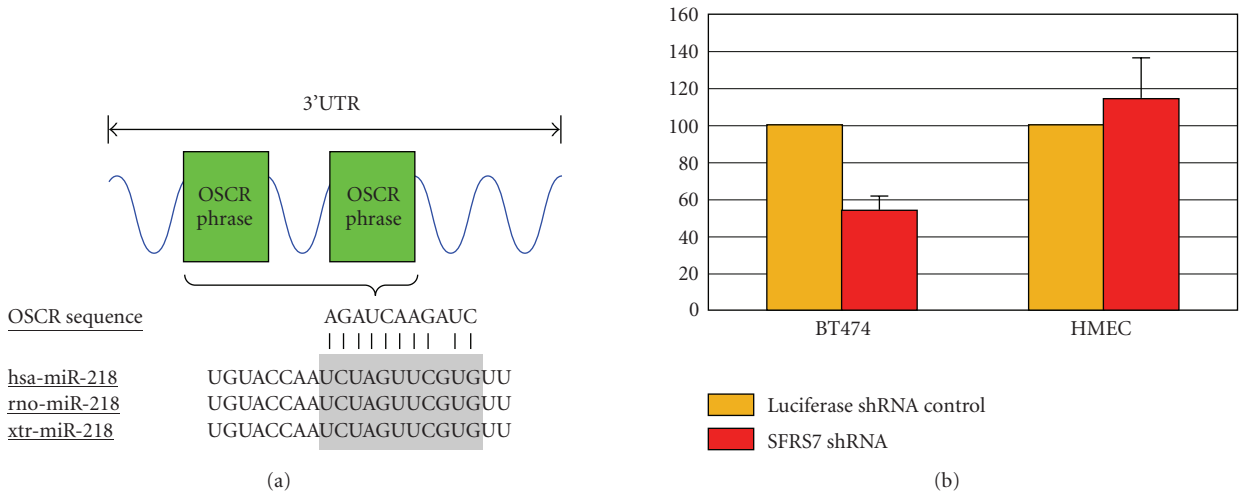
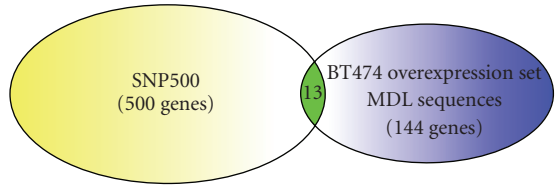


Figure 13: A miRNA target site relevant to breast cancer is identified by OSCR. (a) Proposed interaction between miRNAs (human, rat, frog) and OSCR phrase. (b) Down regulation of the SFRS7 by RNAi specifically inhibits the proliferation of breast cancer cell line BT474 and not normal cells. These miRNAs may be implicated in breast cancer.

single-nucleotide change to a target of one miRNA may yield a target sequence for a distinct miRNA. A report published in 2006 demonstrated this SNP effect in a mammal. The study found that Texel sheep, which are known for their meatiness, possess a mutation in the 3′UTR of the myostatin gene that results in an "illegitimate" interaction of miRNA 1 and 206 with the myostatin mRNA [39]. Mutations that yield such interactions between mutant mRNA and miRNAs are called "Texel-like." The authors performed a preliminary analysis of known human SNPs and their potential for perturbing



(a)

Name	Accession	MDL sequence	Position	SNP
ESR1	NM_000125	GATATGTTTA	4023.5325	4029 T→ C
PTGS2	NM_000963	CAAAATGC	2179, 2717.3097	3103 G→ A
EGFR	NM_005228	TTTTACTTC	4233.4967	4975 C→ T

(b)

Figure 14: MDLcompress directly identifies putative miRNA target sequences that may be implicated in breast cancer. (a) Schematic of overlap between SNP500 database and potential miRNA sequences identified by MDLcompress in the test set. (b) Potential miRNA sites identified by MDLcompress with disease-related polymorphisms identified by SNP analysis. These miRNA targets may be implicated in breast cancer.

binding sites of predicted miRNAs and identified 2490 Texel-like mutations and 483 mutations that potentially result in loss of miRNA binding.

We performed a similar analysis on the 144 overexpressed gene mRNA sequences from the BT474 breast cancer cell line [30, 31] to identify which of these genes possess disease-related Texel-like mutations. By cross-referencing with the SNP500 database [40], SNPs were found in 13 of the 144 overexpressed gene mRNA sequences from the BT474 breast cancer cell line, all in the 3′UTR region. The initial comparison of the 93 MDLcompress code words from the 144 genes discussed previously did not match with any SNP phrases. We then relaxed the strict constraint that a phrase must lead to compression at every step and asked MDLcompress in longest match to identify the top 10 candidates in each gene mRNA sequence that would most likely lead to compression. Strikingly, 3 of these genes-ESR-1, PGTS2, and EGFR-have SNPs in the set of the first 10 code word candidates identified by MDLcompress when run on each these genes respective mRNA sequence (Figure 14). These three sequences were selected out of the 13 because they fulfill the criteria we used for Figure 13(a), that based on sequence analysis (similarity to miRNA sequences and intra- and inter- species sequence conservation); they are putative miRNA targets.

These motifs are localized to the 3′UTR and have not been predicted to interact with any known miRNAs in the literature. Although further validation studies are required, these observations suggest that MDLcompress may be capable of directly identifying potential miRNA target sequences with roles in breast cancer.

Our hypothesis regarding the significance of MDL phrases that are added to the MDLcompress model motivates search of these phrases for SNPs related to cancer. As shown in Figure 10, an SNP identified in PTGS2 gene [40] colocalizes with the MDLcompress-identified phrase caaaatgc in the 3′UTR of PTGS2 and yields a disproportionate change in the descriptive cost of the sequence under the MDLcompress model generated for the original sequence. Altering a single nucleotide typically yields a very small change in descriptive cost, in most cases less than a bit; however, the SNP in the phrase shown in Figure 15 yields a change in descriptive cost on the order of 4 bits, suggesting that this phrase is in fact meaningful. Future work will elaborate on this potential relationship between meaningful phrases identified by MDLcompress and disease, and explore the capability of using MDLcompress models to predict sites where SNPs are especially likely to cause pathology.

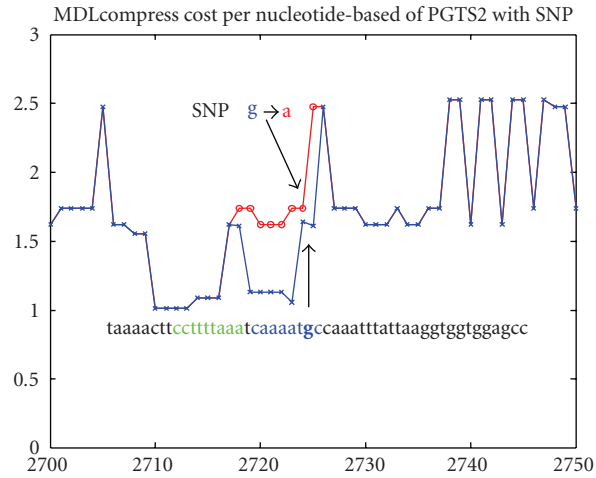


Figure 15: Cost per nucleotide for PTGS2. The blue curve identifies cost per nucleotide of the original sequence based upon an MDLcompress model developed using the total compression heuristic and the first 15 phrases to be selected. The cost per nucleotide under the SNP $g \rightarrow a$ is shown in red.

11. CONCLUSIONS

MDLcompress yields compression of DNA sequences that is superior to any other existing grammar-based coding algorithm. It enables automatic detection of model granularity,

leading to identification of interesting variable-length motifs. These motifs include miRNA target sequences that may play a role in the development of disease, including breast cancer, introducing a novel method of identifying microRNA targets without specifying the sequence (or, in particular, seed) of the microRNA that is supposed to bind them. Additionally, we have used our algorithm here to study SNPs found in overexpressed genes in the breast cancer cell line BT474, and we identified 3 SNPs that may alter the ability of microRNAs to target their sequence neighborhood.

In future work, MDL specificity will be improved through windowing and segmentation, concepts described in Figure 4. Running MDLcompress on consecutive windows of sequence will enable the detection of change points, such as the transition from noncoding to coding sequence, and permit the use of multiple codebooks, enhancing specificity for each region of a gene. For example, the optimal MDL codebook for a coding region is unlikely to be the same as that for a 3′UTR. Applying the same model over an entire gene reduces the effectiveness of the MDL compression algorithm in identifying biologically significant motifs. This improvement of MDLcompress to detect and take advantage of change points will enable the detection of nonadjacent regions of the genome that are similar. The execution time of MDLcompress will be further reduced by means of a novel data structure that augments a suffix tree with counts and pointers, enabling deep recursion of model inference without intractable computation. With this structure, when a phrase is selected for the MDLcompress codebook, simple operations can update the structure to facilitate selection of the next phrase by leveraging known information. The suffix-tree with counts and pointers architecture will enable near-linear time processing of the windowed segments.

ACKNOWLEDGMENTS

This work was funded by the U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick, DM 217-5014 in Grants W81XWH-0-1-0501 (to SE and AT) and W8IWXH-04-1-0474 (to DSC). The content and information do not necessarily reflect the position or policy of the government and no official endorsement should be inferred.

REFERENCES

[1] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello, "Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans," *Nature*, vol. 391, no. 6669, pp. 806–811, 1998.

[2] G. J. Hannon and J. J. Rossi, "Unlocking the potential of the human genome with RNA interference," *Nature*, vol. 431, no. 7006, pp. 371–378, 2004.

[3] A. Kourtidis, C. Eifert, and D. S. Conklin, "RNAi applications in target validation," in *Systems Biology, Applications and Perspectives*, P. Bringmann, E. C. Butcher, G. Parry, and B. Weiss, Eds., vol. 61 of *Ernst Schering Foundation Symposium Proceedings*, pp. 1–21, Springer, New York, NY, USA, 2007.

[4] B. P. Lewis, I.-H. Shih, M. W. Jones-Rhoades, D. P. Bartel, and C. B. Burge, "Prediction of mammalian microRNA targets," *Cell*, vol. 115, no. 7, pp. 787–798, 2003.

[5] B. P. Lewis, C. B. Burge, and D. P. Bartel, "Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets," *Cell*, vol. 120, no. 1, pp. 15–20, 2005.

[6] V. Rusinov, V. Baev, I. N. Minkov, and M. Tabler, "MicroInspector: a web tool for detection of miRNA binding sites in an RNA sequence," *Nucleic Acids Research*, vol. 33, web server issue, pp. W696–W700, 2005.

[7] G. A. Calin, C.-G. Liu, C. Sevignani, et al., "MicroRNA profiling reveals distinct signatures in B cell chronic lymphocytic leukemias," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 32, pp. 11755–11760, 2004.

[8] A. Esquela-Kerscher and F. J. Slack, "Oncomirs—microRNAs with a role in cancer," *Nature Reviews Cancer*, vol. 6, no. 4, pp. 259–269, 2006.

[9] P. Grünwald, I. J. Myung, and M. Pitt, Eds., *Advances in Minimum Description Length: Theory and Applications*, MIT Press, Cambridge, Mass, USA, 2005.

[10] S. C. Evans, *Kolmogorov complexity estimation and application for information system security*, Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, NY, USA, 2003.

[11] S. C. Evans, B. Barnett, S. F. Bush, and G. J. Saulnier, "Minimum description length principles for detection and classification of FTP exploits," in *Proceedings of IEEE Military Communications Conference (MILCOM '04)*, vol. 1, pp. 473–479, Monterey, Calif, USA, October-November 2004.

[12] S. C. Evans, A. Torres, and J. Miller, "MicroRNA target motif detection using OSCR," Tech. Rep. GRC223, GE Research, Niskayuna, NY, USA, 2006.

[13] M. Li and P. Vitányi, *Introduction to Kolmogorov Complexity and Applications*, Springer, New York, NY, USA, 1997.

[14] W. Szpankowski, W. Ren, and L. Szpankowski, "An optimal DNA segmentation based on the MDL principle," *International Journal of Bioinformatics Research and Applications*, vol. 1, no. 1, pp. 3–17, 2005.

[15] I. Tobus, G. Korodi, and J. Rissanen, "DNA sequence compression using the normalized maximum likelihood model for discrete regression," in *Proceedings of Data Compression Conference (DCC '03)*, pp. 253–262, Snowbird, Utah, USA, March 2003.

[16] A. Apostolico and S. Lonardi, "Some theory and practice of greedy off-line textual substitution," in *Proceedings of Data Compression Conference (DCC '98)*, pp. 119–128, Snowbird, Utah, USA, March 1998.

[17] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical structure in sequences: a linear-time algorithm," *Journal of Artificial Intelligence Research*, vol. 7, pp. 67–82, 1997.

[18] N. Cherniavsky and R. Lander, "Grammar-based compression of DNA sequences," in *DIMACS Working Group on The Burrows—Wheeler Transform*, Piscataway, NJ, USA, August 2004.

[19] X. Chen, M. Li, B. Ma, and J. Tromp, "DNACompress: fast and effective DNA sequence compression," *Bioinformatics*, vol. 18, no. 12, pp. 1696–1698, 2002.

[20] B. Behzadi and F. Le Fessant, "DNA compression challenge revisited: a dynamic programming approach," in *The 16th Annual Symposium on Combinatorial Pattern Matching (CPM '05)*, vol. 3537 of *Lecture Notes in Computer Science*, pp. 190–200, Jeju Island, Korea, 2005.

[21] S. C. Evans, T. S. Markham, A. Torres, A. Kourtidis, and D. Conklin, "An improved minimum description length learning algorithm for nucleotide sequence analysis," in *Proceedings of IEEE 40th Asilomar Conference on Signals, Systems and*

Computers (ACSSC '06), pp. 1843–1850, Pacific Grove, Calif, USA, October-November 2006.

[22] P. Gács, J. T. Tromp, and P. M. B. Vitányi, "Algorithmic statistics," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2443–2463, 2001.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.

[24] E. C. Lai, "MicroRNAs are complementary to 3′ UTR sequence motifs that mediate negative post-transcriptional regulation," *Nature Genetics*, vol. 30, no. 4, pp. 363–364, 2002.

[25] E. C. Lai, B. Tam, and G. M. Rubin, "Pervasive regulation of *Drosophila* Notch target genes by GY-box-, Brd-box-, and K-box-class microRNAs," *Genes & Development*, vol. 19, no. 9, pp. 1067–1080, 2005.

[26] J. G. Doench and P. A. Sharp, "Specificity of microRNA target selection in translational repression," *Genes & Development*, vol. 18, no. 5, pp. 504–511, 2004.

[27] J. Brennecke, A. Stark, R. B. Russell, and S. M. Cohen, "Principles of microRNA-target recognition," *PLoS Biology*, vol. 3, no. 3, p. e85, 2005.

[28] S. C. Evans, G. J. Saulnier, and S. F. Bush, "A new universal two part code for estimation of string kolmogorov complexity and algorithmic minimum sufficient statistic," in *DIMACS Workshop on Complexity and Inference*, Piscataway, NJ, USA, June 2003.

[29] P. M. Voorhoeve, C. le Sage, M. Schrier, et al., "A genetic screen implicates miRNA-372 and miRNA-373 as oncogenes in testicular germ cell tumors," *Cell*, vol. 124, no. 6, pp. 1169–1181, 2006.

[30] A. Mackay, C. Jones, T. Dexter, et al., "cDNA microarray analysis of genes associated with *ERBB2* (HER2/*neu*) overexpression in human mammary luminal epithelial cells," *Oncogene*, vol. 22, no. 17, pp. 2680–2688, 2003.

[31] F. Bertucci, N. Borie, C. Ginestier, et al., "Identification and validation of an *ERBB2* gene expression signature in breast cancers," *Oncogene*, vol. 23, no. 14, pp. 2564–2575, 2004.

[32] L. P. Lim, N. C. Lau, P. Garrett-Engele, et al., "Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs," *Nature*, vol. 433, no. 7027, pp. 769–773, 2005.

[33] S. F. Altschul, T. L. Madden, A. A. Schäffer, et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[34] F. Mignone, G. Grillo, F. Licciulli, et al., "UTRdb and UTRsite: a collection of sequences and regulatory motifs of the untranslated regions of eukaryotic mRNAs," *Nucleic Acids Research*, vol. 33, database issue, pp. D141–D146, 2005.

[35] http://microrna.sanger.ac.uk/sequences/index.shtml.

[36] S. Griffiths-Jones, R. J. Grocock, S. van Dongen, A. Bateman, and A. J. Enright, "miRBase: microRNA sequences, targets and gene nomenclature," *Nucleic Acids Research*, vol. 34, database issue, pp. D140–D144, 2006.

[37] X. Huang, R. C. Hardison, and W. Miller, "A space-efficient algorithm for local similarities," *Computer Applications in the Biosciences*, vol. 6, no. 4, pp. 373–381, 1990.

[38] P. J. Paddison, J. M. Silva, D. S. Conklin, et al., "A resource for large-scale RNA-interference-based screens in mammals," *Nature*, vol. 428, no. 6981, pp. 427–431, 2004.

[39] A. Clop, F. Marcq, H. Takeda, et al., "A mutation creating a potential illegitimate microRNA target site in the myostatin gene affects muscularity in sheep," *Nature Genetics*, vol. 38, no. 7, pp. 813–818, 2006.

[40] http://snp500cancer.nci.nih.gov/.