ENHANCING GENE DETECTION WITH COMPUTER GENERATED INTERGENIC REGIONS

Juan Caballero and Gustavo Glusman

Institute for Systems Biology,
1441 N 34th Street, 98103, Seattle, WA, USA,
jcaballero@systemsbiology.org, gustavo@systemsbiology.org

ABSTRACT

Coding and non-coding gene prediction is still a challenge. Diverse computer-based tools have been created to screen sequences using elaborate strategies for gene prediction. Many of these implement various statistical tests to measure the plausibility of the prediction but until now, a comprehensive negative control did not exist.
We developed an algorithm that generates sequences with characteristics of the intergenic regions of a genome, including nucleotide composition and typical inserted elements like interspersed repeats, low complexity sequences and pseudogenes. We also challenged some gene prediction programs to compare the artificial sequences with real intergenic regions.

1. INTRODUCTION

New advances in DNA sequencing are increasing our opportunity to know the complete genome of many organisms, including human, and projects like the "$1,000 genome" [1] motivate the development of faster and cheaper methodologies.

Gene detection programs are an essential tool to discover functional regions in the screened sequences. Modern gene prediction tools rely on three basic concepts: (1) modeling of gene structure [2], (2) recognition of sequence similarity [2] and (3) detection of signatures of transcription, accumulated over evolutionary time [3]. These programs incorporate their own scoring system or statistical tests for the plausibility of gene predictions, but none filter the predictions based on negative control tests.

We present here a new algorithm that produces artificial non-coding sequences of any desired size. The artificial sequences are designed to be indistinguishable from real non-functional sequences.

We implemented this algorithm in the Perl scripting language, for multiple platform compatibility. The source code and the models used in this study are available upon request to the authors.

2. MATERIALS AND METHODS

2.1. Genomic information

The human genome (build hg18) and the annotation database where obtained from UCSC Genome Browser (http://genome.ucsc.edu).

2.2. Intergenic profiles

We removed from the reference genome sequence (3.1 Gbp) all regions annotated as known genes, non-coding RNA, pseudogenes, interspersed repeats or simple repeats. After removing all these, ~700 Mbp remained (24% of the genome). For each sequence we computed the *k*-mer composition (k = <1, 2, 3, 4, 5, 6>) on the alphabet A = <A, G, C, T> in non-overlapping windows w = <20, 50, 100, 200, 500, 1000> and classified each window by GC content gc = <0-5, 5-10, 10-15, …, 95-100>. We also computed the frequency of GC content transitions between windows.

2.3. Sequence generation

The method to generate a new artificial sequence consists of two steps: (1) generate a base sequence, and (2) insert into it the expected number of repeats and pseudogenes based on genome distributions.

The new sequence is created with the profiles of a predefined *k*-mers composition allowing changes of GC content after each completed windows. Every step include a probability of GC transition based in the genome profile.
Repeats are then inserted into the base sequence at random locations and orientation. The most diverged repeats are inserted first, to prevent the insertion of older repeats into younger repeats, using the number of mutations as an age parameter. Finally pseudogenes are inserted, also in random position/orientation.

2.4. Elements for insertion

We classified the repeats into two classes: (1) simple or low complexity repeats and (2) interspersed repeats. The first class is reported in the UCSC Genome Browser database as "simple repeats table", which includes the monomer sequence and the number of copies. Our algorithm takes this information and recreate the sequence by expanding the monomer. For interspersed repeats we used the alignment output of full genome masking with RepeatMasker 3.2.6 [4]. For each repeat, we extracted the fragment coordinates and mutation rates (transitions, transversions, insertions and deletions) derived from the consensus reported in RepBase rel. 20071204 [5]. We used this information to artificially evolve repeats by extracting the corresponding fragment of the consensus and randomly altering it using its own mutation rate.

2.5. Sequence validation

Real genomic sequences have certain constrains in composition and complexity, which are captured by our training process. Since our artificial sequences are by definition unique due the random generation process, we compared them with real sequences using alignment-free methods. We evaluated the sequences in terms of bases composition, dimer skew and inner complexity comparing: (1) real 100 kb long genomic sequences from intergenic regions, (2) the same intergenic sequences permuted independently using the dinucleotide composition conservation with the Fisher-Yates shuffle algorithm, (3) an equal number of similarly sized artificial sequences generated with dimer models and window size of 100 bases, and (4) an equal number of randomly chosen fragments from human chromosome 3, which may include some real genes (~1 gene / 100 kb according to UCSC database).

2.5.1. Sequence composition

Composition quantifications for each sequence where calculated in terms of:

- G+C percent

$$gc = (n_G + n_C)/N \quad (1)$$

- G+C skew

$$gcs = (n_G - n_C)/(n_G + n_C) \quad (2)$$

- A+T skew

$$ats = (n_A - n_T)/(n_A + n_T) \quad (3)$$

- G+A skew

$$gas = (n_G - n_T + n_A - n_C)/N \quad (4)$$

- G+T skew

$$gts = (n_G + n_T - n_A - n_C)/N \quad (5)$$

- CpG ratio

$$cpg = n_{CG}/(n_G \cdot n_C) \quad (6)$$

where n_i represent the number of occurrences of i element in the sequence and N is the total length of the sequence.

2.5.2. Sequence complexity

For complexity measure we use the following methods:

- Complexity by Wootton & Federhen [6].

$$cwf = \frac{1}{N} . \log_4 \frac{(N!)}{(\prod_{i=1}^{4} n_i!)} \quad (7)$$

- Complexity entropy [7].

$$ce = -\sum_{i=1}^{4} (\frac{n_i}{N}) . \log_4 \frac{n_i}{N} \quad (8)$$

- Complexity entropy of Markov model [7].

$$cm = -\sum_{i=1}^{4^m} M . \log_{(4^m)} M \quad (9)$$

where $M = \frac{m_i}{(N - m - 1)}$, m is the word size and m_i: total words for i^{th} word.

- Linguistic complexity [8].

$$cl = (\sum_{j=1}^{m} V_j)/(\sum_{j=1}^{m} V_{MAXj}) \quad (10)$$

where $V_{MAXj} = min(m^j, N - j - 1)$, V_j total unique words j size, V_{maxj} is the maximal number of possible word with j size and m is the maximal word size.

- Bit compression ratio [9].

$$cz = Z_{(N)}/N \quad (11)$$

where: $Z_{(N)}$ = compression of N with Zlib.

2.5.3. Multiple factor analysis

We built a matrix with the calculated values for sequence composition and complexity using masked repeats and full length windows. The complexity scores in equations 9, 10 and 11 were calculated with word lengths (m) from monomers to hexamers. The matrix was analyzed using principal component analysis in MatLab (http://www.mathworks.com/) to scale the multiple values into a 2D plot and visually determine the presence of biased groups.

2.6. Gene prediction

2.6.1. Coding genes

We used Genscan 1.0 [10], GlimmerHMM 3.0.1 [11], Twinscan 3.5 [12], and Augustus 2.0.3 [13] as *ab initio* gene prediction programs, all programs were used with a trained model for detection of human genes (provided by the authors in their respective programs), to scan 1,000 repeats-masked sequences for putative coding fragments.

2.6.2. Non-coding genes

The Infernal 1.0 [14] program was used to predict putative non-coding RNAs. We manually selected twelve models from Rfam 9.1 [15], the public database of non-coding RNA, to scan 100 repeat-masked sequences.

3. RESULTS

3.1. Sequence generation

We evaluated the parameters in terms of k-mer model and window size to produce artificial sequences similar properties to those of real intergenic sequences. While the composition and complexity tests cannot discriminate between real and artificial sequences (Figure 1), the

models using *k*-mer profiles > 2 produces more realistic sequences than mononucleotide profiles. In the subsequent analysis we used artificial sequences generated with a dimers profile (*k*=2). We observed consistently results when using window sizes of 20-500 bases (data not shown).

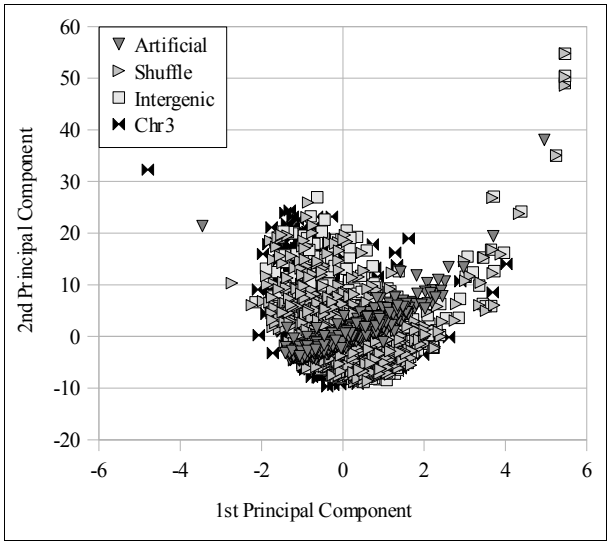

Figure 1. Principal components analysis of composition and complexity measures for the different data sets.

3.2. Coding gene predictions

All the programs tested predicted coding regions in the artificial sequences (Figure 2). Some programs reported complete genes including transcription signals, exons and introns. The longest predicted genes spanned of ~2-3 kb. We used NCBI's blast to search for similarities with the database of non- redundant peptides (nr). As expected, this comparison found no significant hits (e-value < 0.001) for all the predicted protein sequences.

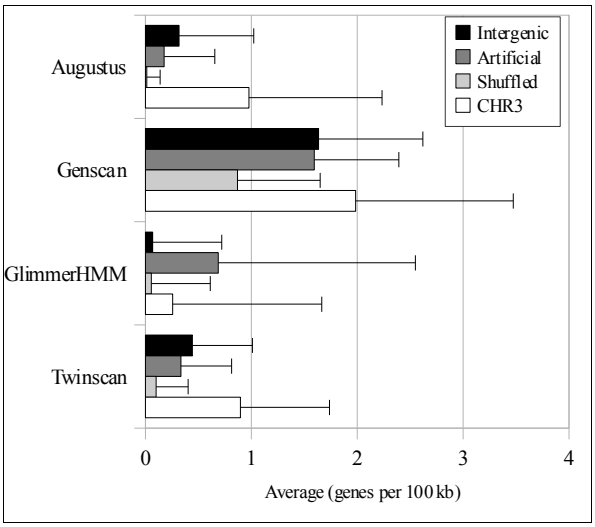

Figure 2. Coding-gene predictions rates by various algorithms.

We observed a similar number of predictions for the artificial and real intergenic sequences, while shuffled intergenic sequences had lower values. A comparison of Genscan overall scores and predicted peptides showed a similar pattern in real intergenic and artificial sequences, while the shuffled intergenic sequences obtained lower scores and length (Figure 3).

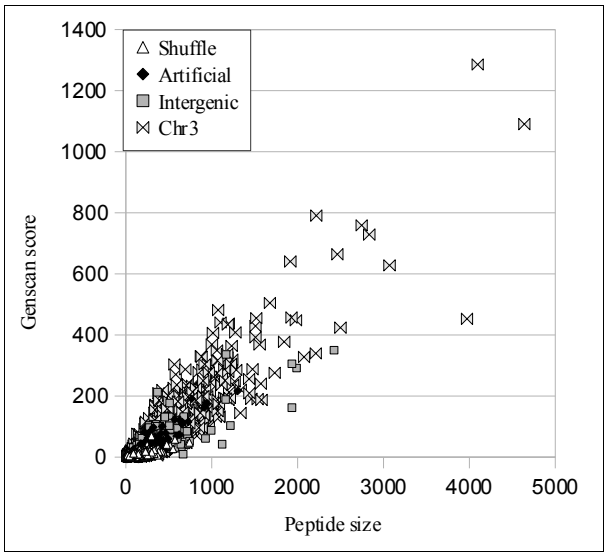

Figure 3. Genscan predictions.

3.3. Non-coding gene predictions

A similar screen with Infernal and Rfam models predicted non-coding RNAs in the artificial sequences. We manually selected ten of the 1,372 models available in Rfam due the prohibitively computational time for some models (Figure 4). Infernal predicted non-coding RNAs in the artificial and real intergenic sequences, but none in the shuffled intergenic sequences. The scores and the E-value calculated by Infernal in the predictions have the same pattern in the intergenic and artificial sequences (Figure 5).

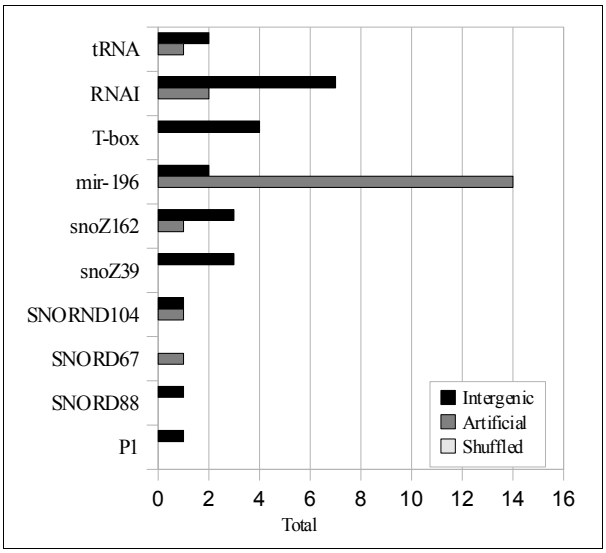

Figure 4. Infernal predictions for some ncRNA models.

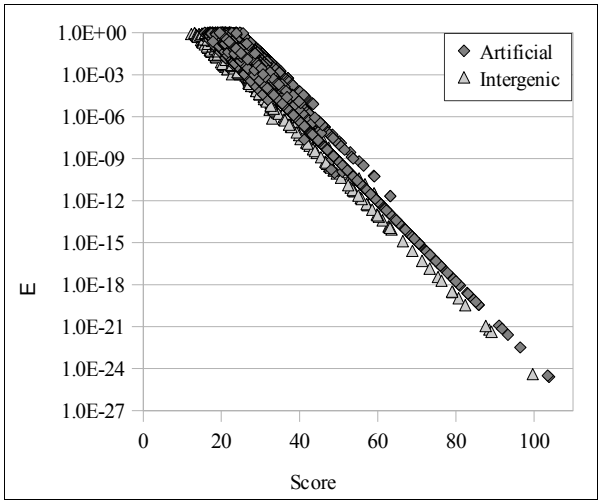

Figure 5. Infernal predictions scores and E-value.

4. CONCLUSION

Computer generated sequences are difficult to produce with the same properties of real genomic sequences, and efforts like ours are needed to correctly model the DNA in a genome. Screening genomic sequences for functional parts requires statistical criteria for plausibility. In this sense, a null control is always required.

Our results suggest, that the dinucleotide shuffle methods are not good enough to serve as negative control in gene prediction. Our artificial sequences reveal that composition artifacts are much more common than thought, and the shuffling method does not reproduce this effect.

The scores reported by the gene prediction programs can be compared to a distribution of scores observed on artificial sequences. A p-value can be estimated from this distribution, reducing the false-positive rate by using a more realistic threshold.

5. FUTURE WORK

We plan to continue improving our algorithm, to obtain more realistic sequences, and to apply the method to new applications beyond gene predictions, like full genome simulations, repeats evolution and movement inside a genome, detection of other genome elements like *cis*-regulatory elements and probing other prediction systems.

6. ACKNOWLEDGMENTS

This work is supported by NIH grant R01 GM081083-01A1 “Further development of the FEAST software, and its use for novel gene predictions.”

7. REFERENCES

[1] E.R. Mardis, “The impact of the next-generation sequencing technologies on genetics,” *Trends Genet,* vol. 3, pp. 133-141, 2008.

[2] M.R. Brent, “How does eukaryotic gene prediction work?,” Nat Biotechnol, vol. 25(8), pp. 883-885, 2007.

[3] G. Glusman, S. Qin, M.R. El-Gewely, A.F. Siegel, J.C. Roach, L. Hood, A.F.A. Smit, “A third approach to gene prediction suggests thousands of additional human transcribed regions”, PLoS Comp Biol, vol. 2, pp. 160-173, 2006.

[4] A.F.A. Smit, R. Hubley and P. Green, “RepeatMasker Open-3.0”, 1996-2004, http://www.repeatmasker.org/.

[5] J. Jurka, V.V. Kapitonov, A. Pavlicek, P. Klonowski, O. Kohany and J. Walichiewicz, “Repbase Update, a database of eukaryotic repetitive elements”, Cytogentic and Genome Res, vol. 110, pp. 462-467, 2005.

[6] J.C Wootton and S. Federhen, “Analysis of compositionally biased regions in sequence databases”, Methods Enzymol., vol. 266, pp. 554–557, 1996.

[7] Y.L. Orlov and V.N. Potapov, ”Complexity: an internet resource for analysis of DNA sequence complexity”, Nuc Ac Res, vol. 32, pp. W628–W633, 2004.

[8] O.G. Troyanskaya, O. Arbell, Y. Koren, G.M. Landau, and A. Bolshoy, “Sequence complexity profiles of prokaryotic genomic sequences: a fast algorithm for calculating linguistic complexity”, Bioinformatics, vol. 18, pp. 679–688, 2002.

[9] M.G Sadovsky, J.A. Putintseva, A.S. Shchepanovsky, “Genes, information and sense: complexity and knowledge retrieval”, Theory Biosci, vol. 127, pp 69-78, 2008.

[10] C. Burge and S. Karlin, “Prediction of complete gene structures in human genomic DNA”, J. Mol. Biol., vol. 268, pp. 78-94, 1997.

[11] W.H. Majoros, M. Pertea and S.L. Salzberg, “TigrScan and GlimmerHMM: two open-source ab initio eukaryotic gene-finders”, Bioinformatics, vol. 20, pp. 2878-2879, 2004.

[12] W.H. Majoros, M. Pertea, A.L. Delcher and S.L. Salzberg, ”Efficient decoding algorithms for generalized hidden Markov model gene finders”, BMC Bioinformatics, vol. 6, p. 16, 2005.

[13] M. Stanke and B. Morgenstern, "AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints", Nuc Ac Res, vol. 33, pp. W465-W467, 2005.

[14] SR Eddy, "A memory efficient dynamic programming algorithm for optimal structural alignment of a sequence to an RNA secondary structure", BMC Bioinformatics, vol. 3, pp. 18, 2002.

[15] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S.R. Eddy, “Rfam: an RNA family database”, Nuc Ac Res, vol. 31(1), pp. 439-441, 2003.