M Are Better Than One: An Ensemble Method For Genomic Island Prediction

Dongsheng Che, Mohammad Shabbir Hasan, Han Wang, John Fazekas
Department of Computer Science
East Stroudsburg University
East Stroudsburg, PA, 18301, USA
dche@po-box.esu.edu

Bernard Chen
Department of Computer Science
University of Central Arkansas
Conway, AR, 72035, USA
bchen@uca.edu

Xiuping Tao
Department of Chemistry and Physics
Winston-Salem State University,
Winston-Salem, NC 27110.
taoxi@wssu.edu

***Abstract*—A Genomic island (GI) is a part of genomic sequence that originally transferred from other organisms, and now it is stabilized into the host genome. The detection of GIs is significant to evolutionary studies as well as biomedical research. Different computational methods for GI detection often lead to different predicted GIs, raising the issues of what predicted GIs are true GIs. In this paper, we propose an ensemble learning approach that uses the prediction results of multiple GI tools, filters out noisy predictions, and generates consensus prediction results. The performance evaluation test has shown that our ensemble approach was more accurate than any other single GI prediction program. The coefficient correlation analysis has also shown that our approach was more correlated to other programs overall, strongly suggesting the reliability of our ensemble algorithm for GI prediction.**

Keywords-ensemble method; genomic island; prokaryote; sequence analysis.

I. INTRODUCTION

Genomic islands (GIs) are parts of genomic regions that have the origin of horizontal gene transfer. Because of their origin, GIs can be found different from other parts of genomic sequences in their sequence composition, such as GC content, codon usage, and *k*-mer nucleotide frequency. GIs often contain mobile genes, such as integrase gene and transposase gene. In addition, some genomic islands are flanked by transfer RNA (t-RNA) genes.

The identification of genomic islands is significant to biomedical research and pharmaceutical companies. For instance, the detection of pathogenic GIs can help microbiologists to understand the mechanisms of pathogenicity of the organism, and thus promote pharmaceutical companies to design related vaccines and antibiotics. On the other hand, some other GIs of bacterial genomes contain second metabolite associated genes, such as POLYKETIDES genes. Therefore, identifying such GIs can help researchers understand the machinery of metabolisms, and promotes pharmaceutical companies to produce natural products of medicines at the large scale.

Computational methods have been developed for GI detection. Existing prediction tools include AlienHunter [1], Centroid [2], COLOMBO SIGI-HMM [3], IslandPath [4], INDeGenIUS [5], and PAI-IDA [6]. All of these prediction tools use one or multiple features of sequence composition, mobile genes, or tRNA gens. For example, AlienHunter uses the variable-length *k*-mers to measure sequence compositional bias, while COLOMBO SIGI-HMM measures codon usage bias as a genome sequence composition signature. The prediction results of GIs using different approaches were different, making it difficult for users to decide which predicted GIs are truly GIs. IslandViewer [7] provides the interface for all the predicted results of three programs, COLOMBO SIGI-HMM, IslandPath and IslandPick [8]. The system itself does not decide which predicted GIs are true GIs, leaving the users to make decisions of which predicted GIs are true GIs.

It is not uncommon that different computational approaches generate different prediction results in the field of bioinformatics. For instance, different motif-finding programs usually generate different predicted motifs [9]. To solve this type of problem in motif-finding, researchers have used ensemble learning approaches to combine the prediction results of multiple programs. Such ensemble-based motif-finding approaches include BEST [10], EMD [9], and MotifVoter [11]. The prediction results using ensemble-based approaches could be improved in general from these studies.

In this paper, we propose our ensemble-based method for genomic island prediction. We use the prediction results of five GI tools, AlienHunter COLOMBO SIGI-HMM, IslandPath, INDeGenIUS, and PAI-IDA, make the votes on predicted results, and generate final consensus GI regions using our ensemble algorithm. Both coefficient correlation and prediction accuracy analysis suggested the reliability of our predicted GIs. Therefore, we believe the usefulness of our ensemble-based GI tool for the future genome annotation. This paper is organized as follows: Section 2 describes our computational framework as well as ensemble algorithm. Section 3 shows the performance results of our ensemble method. We conclude in Section 4, with the discussion of our future work.

II. METHODS

A. Computational Framework

The computational framework for GI prediction is as follows:

This research was partially supported by President Research Fund, and Faculty Professional Development & Research (FDR) major grant at East Stroudsburg University of Pennsylvania.

978-1-61284-101-4/12/$26.00 ©2012 IEEE

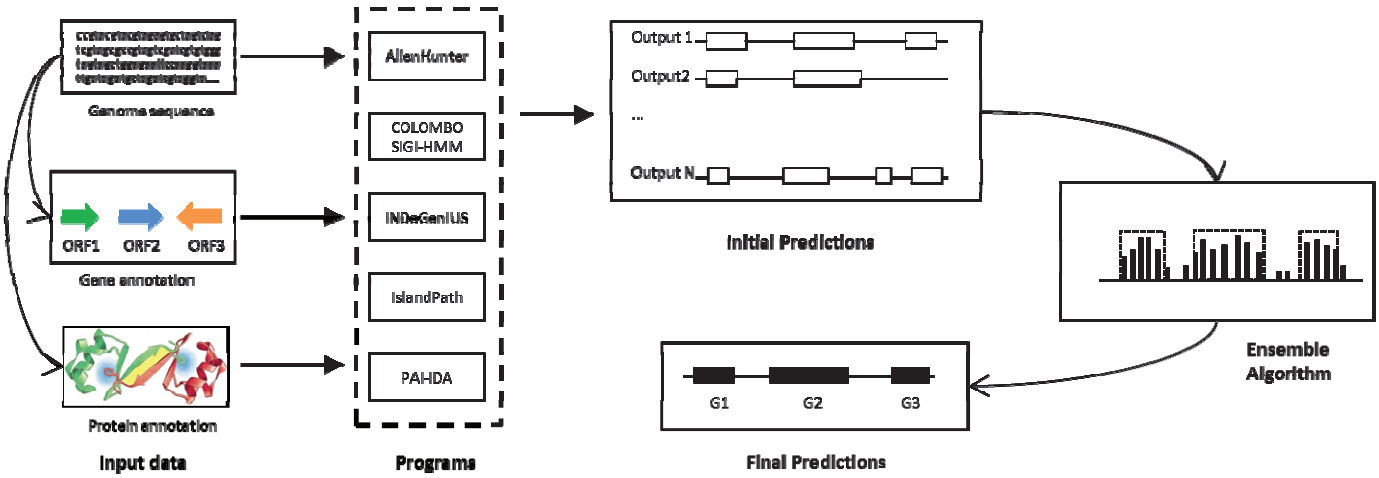

Figure 1. Computational framework for GI prediction

TABLE I. SUMMARY ABOUT THE OPERATING PRINCIPLES, DATA REQUIREMENT, URLS OF COMPONENT ALGORITHMS

Program	Operating Principle	Input Files	URL	Ref
AlienHunter	Uses variable-length *k*-mers (*i.e.,* IVOMs) to measure sequence compositional bias.	FNA	http://www.sanger.ac.uk/resources/software/alien_hunter/	[1]
COLOMBO SIGI-HMM	Measures codon usage bias as a genome sequence composition signature.	GBK	http://www.tcs.informatik.uni-goettingen.de/colombo-sigihmm	[3]
INDeGenIUS	Splits the genome into regions of a certain size, and measure sequence composition bias with different *k*-mers.	FNA	Available upon request	[2]
IslandPath	Measures G+C content, dinucleotide bias, the presence of RNA and mobility gene	FAA, FFN, PTT	http://www.pathogenomics.sfu.ca/islandpath	[4]
PAI-IDA	Uses GC content, dinucleotide frequency and codon usage.	GBK	http://compbio.sibsnet.org/projects/pai-ida/	[6]

1) *Collect genomic datasets.* The datasets used for GI prediction include the genomic sequence of the query genome, gene annotations, and protein annotations.

2) *Run GI prediction tools*. Run each of GI software tools to generate initial GI locations.

3) *Run our ensemble algorithm to produce final GI locations*. The predicted GI locations by multiple tools will be collected, filtered and processed using our ensemble algorithm.

The schematic view of the flowchart of the framework is shown in Figure 1. The details of the GI tools and our ensemble algorithm are described in the following subsections.

B. Component Programs

The GI prediction tools embedded in our computational frame include AlienHunter, COLOMBO SIGI-HMM, IslandPath, INDeGenIUS, and PAI-IDA. We should point out that some GI programs were not included. For instance, Centroid was not included since INDeGenIUS is an improved version of Centroid. The comparative genomics based GI tools were not included either since they have the requirements of multiple reference genomes, limiting the usage of GI prediction on any query genome. The software we are developing can be applied to any query genome.

The operating principle of each GI tool is summarized in Table 1. Each program needs different genomic data. For instance, AlienHunter only needs the genomic sequence while IslandPath needs genomic sequence, gene annotations, and corresponding protein annotations.

The Genomic sequences can be obtained from the National Center for Biotechnology Information (NCBI) FTP server (ftp://ftp.ncbi.nih.gov/genomes/Bacteria). The detailed file formats required for each program are listed in Table 1. The predicted GI location results are collected and used in our ensemble algorithm.

C. Ensemble Algorithm

The basic idea of our ensemble algorithm is to make votes based on all predicted GI locations, and then pick the consensus GI regions based on a threshold value. The technical details of the ensemble algorithm are discussed as follows:

First of all, genomic islands are typically long, spanning from several *kb* to several hundred *kb*. If we use each predicted GI region as a comparison or voting unit, there will be no exact match between GI locations predicted by multiple tools. On the other hand, if we treat each nucleotide as a unit for comparison, there will be so many nucleotides for comparison, making the computation time too long. To this end, we choose each gene in the genome as a comparison and voting unit. For any gene g_i within the

genome $G = \{g_1, g_2, \ldots g_i, \ldots g_m\}$, we define $|g_i|$ to be the number of tools that predict g_i to be within GI locations, and accordingly, we have a string of votes, *i.e.*, $GV = \{|g_1|\ |g_2| \ldots |g_i| \ldots |g_m|\}$.

One naive scheme to obtain final GIs is to separate the string GV into a set of substrings such that each substring contains a number of contiguous genes, where all genes' $|g_.|$ are greater than a threshold value. We consider the region covering the genes in a substring to be a *GI region*, and the remaining part of the genome to be *non-GI regions*. One of the potential problems is that this scheme relies on the votes of each gene, thus any under-predicted gene (i.e., does not predict it as a GI gene) could result in the split of GIs, and consequently lead to too many small sized, physically close GIs (*e.g.*, 2-3 kb). These small sized GIs should be part of a big GI, since GIs tend to be large, usually covering dozens and even several hundred genes. One alternative solution is to decrease the threshold value, so that neighboring separate GI regions can be formed into a big GI, but that will cause another problem, *i.e.*, including non-GI regions into GIs.

To avoid the problem mentioned above, we use the overall score of a genomic region, rather than individual votes, to measure the GI qualification of the region. To do so, we separate the string GV into a set of substrings $\{GI'_1, GI'_2, \ldots, GI'_i, \ldots, GI'_n\}$, with each substring containing a number of genes, and all genes' $|g_.| > 0$. Now we measure whether any adjacent pair (GI'_i, GI'_{i+1}) can be merged into a big GI based on the overall score of all genes from the first gene in GI'_i to the last gene in GI'_{i+1}. If the overall score meets with the threshold value, the two regions should be merged. Otherwise, it will not be merged. The details of our ensemble algorithm for GI prediction are shown in Figure 2. Our ensemble algorithm, EGID, was implemented in Java, and it could be executed in Linux operating systems.

Algorithm Detect-GIs ($GI_1, GI_2, .., GI_m$)
Input: *m* sets of GI locations by *m* GI tools
Output: a final set of GI locations
Steps:

1. For any gene g_i in the genome,
   $|g_i| \leftarrow$ the number of tools that predict g_i within GI locations;
2. Let $GV = \{GI'_1, GI'_2, \ldots, GI'_i, \ldots, GI'_n\}$, and
3. Let $GI'_i = (g_{i,1}\, g_{i,2} \ldots g_{i,m})$, where all $|g_{i,.}| > 0$;
4. Let $GI'_{i+1} = (g_{i+1,1}\, g_{i+1,2} \ldots g_{i+1,n})$, where all $|g_{i+1,.}| > 0$;
5. Let k = the number of genes between GI'_i and GI'_{i+1};
6. Compute s = Score(GI'_i, GI'_{i+1}, k);
7. If $s > threshold$
   Merge the region between GI'_i and GI'_{i+1}, i.e., $GI'_{new} = (g_{i,1}\, g_{i,2} \ldots g_{i+1,n})$;
8. Repeat the merging process (Step 3-6) using an adjacent GI' pair;
9. Post-process GIs.

End of Algorithm

Figure 2: Ensemble algorithm for GI detection

D. Performance evaluation

We used genomic island datasets picked by Islandpick [8] as the benchmark to evaluate our approach. The benchmark datasets contain 771 positives and 3770 negatives from 118 genomes. Let true positives (TP) be the nucleotides in the positive benchmark dataset predicted to be genomic islands; true negatives (TN) be the nucleotides in the negative benchmark dataset predicted to be non-genomic islands; false positives (FP) be the nucleotides in the negative benchmark dataset predicted to genomic islands; and false negatives (FN) be the nucleotides within the positive benchmark dataset not predicted to be genomic islands. We measure four metrics as follows,

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (1)$$

$$\text{Specificity} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{nPc} = \frac{TP}{TP + FP + FN} \quad (3)$$

$$\text{F} - \text{measure} = \frac{2 * Sensitivity * Specificity}{Sensitivity + Specificity} \quad (4)$$

We also measure correlation coefficient between any pair of tools. Particularly, we use the predicted GI results of one tool as the benchmark dataset to measure TP, TN, FP, and FP of another tool, and we define the correlation coefficient (CC) as:

$$\text{CC} = \frac{nTP * nTN - nFN * nFP}{\sqrt{(nFP + nFN)(nTN + nFP)(nTP + nFP)(nTN + nFN)}} \quad (5)$$

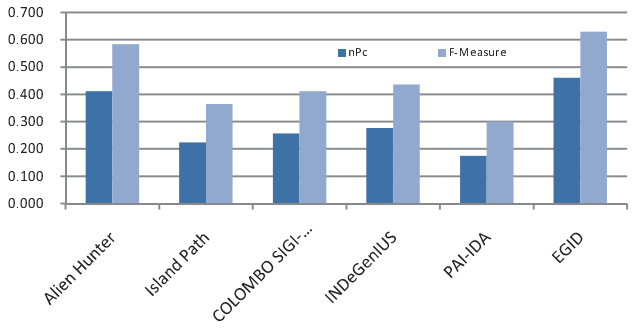

Figure 3. Performance comparison of genomic island programs

III. Experimental Results and Discussions

A. Prediction accuracy analysis

To test the performance of our ensemble based GI prediction method, we have predicted GIs on 118 genomes corresponding to those used in the benchmark GI datasets. We then measured sensitivity, specificity, nucleotide level performance coefficient (nPc), and F-measure. For the comparison purpose, we also measured the prediction accuracies of five other tools used in our ensemble method. Figure 3 shows the values of nPc and F-measure for each tool, and it is easy to see our ensemble approach tops all

other tools based on these two measurements, indicating the performance improvement by our ensemble method.

B. *Correlation coefficient analysis*

We also measured the correlation coefficient between any pair of tools, as shown in Table 2. In general, a high correlation coefficient indicates that two tools predict similar GIs. Overall, the correlation coefficient values between our tool and others are higher than those of other pairs, suggesting that our predicted results were based on the consensus of other prediction tools.

TABLE II. CORRELATION COEFFICIENT FOR ALL PAIRS OF TOOLS

Tools	Alien Hunter	Island Path	COLOMBO SIGI-HMM	INDeGenIUS	PAI-IDA	EGID
Alien Hunter	1.000	0.264	0.317	0.452	0.247	0.573
Island Path	0.264	1.000	0.220	0.268	0.218	0.398
COLOMBO SIGI-HMM	0.317	0.220	1.000	0.255	0.198	0.368
INDeGenIUS	0.452	0.268	0.255	1.000	0.355	0.512
PAI-IDA	0.247	0.218	0.198	0.355	1.000	0.353
EGID	0.573	0.398	0.368	0.512	0.353	1.000
Average	0.475	0.395	0.393	0.474	0.395	0.534

C. *An example: predicted GIs in Escherichia coli E24377A*

We used the genome of *E. coli* E24377A as an example to compare the predicted GIs of our tool with those of five other tools. Figure 4 shows the graphic representation of predicted GIs of all tools. In general, our tool picks the consensus GIs predicted by other tools. For instance, AlienHunter predicted two GIs, with the locations of (4,842,500 -> 4,860,000) and (4,865,000 -> 4,905,000), COLOMBO SIGI-HMM predicted four locations, (4,838,828 -> 4,840,287), (4,843,821 -> 4,848,206), (4,876,346 -> 4,879,082), (4,890,592 -> 4,897,097), INDeGenIUS predicted two GIs, with the locations of (4,870,001 -> 4,880,000), (4,890,001 -> 4,900,000), IslandPath predicted two GIs, with the locations of (4,839,010 -> 4,858,342), (4,868,452 -> 4,886,769). PAI-IDA does not predict any GI in this region. Based on these prediction results, EGID predicted a GI from genome location of 4,838,828 to 4,905,291, covering 82 genes (See Figure 4 label 1).

IV. CONCLUSION AND FUTURE WORK

We have implemented an ensemble-based approach for GI prediction. Pairwise coefficient correlation analysis of GI tools have shown that our GI tool is more correlated to other tools overall when compared with other pairs of tools. In addition, our GI tool is more accurate than any other tools when using the benchmark dataset generated by IslandPick.

Since our software tool uses five existing tools, with each of tools requiring specific genome data and file formats. It is required that users download these file formats through NCBI web server first before running our software tool, thus making it inconvenient for users to do it manually. In our future work, we will develop a GUI interface that provides the functionality of automatic downloading such genome files through NCBI server. The GUI will also automatically set up the environments for users to run the whole computational framework and generate final GI locations.

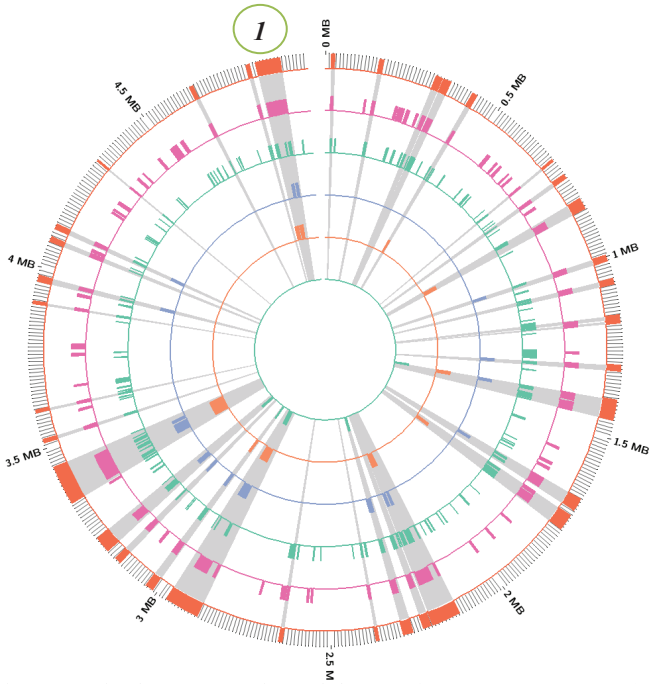

Figure 4. Circular representations of the *E. coli* E24377A (NC 009801) showing predicted GIs, with each circle predicted by each program. The predicted GIs from the outer to the inner circle are EGID, AlienHunter, COLOMBO, INDeGenIUS, Island-Path, and PAI-IDA. The shaded parts show the predicted GIs by EGID, and evidenced GIs by other programs.

REFERENCES

[1] G. S. Vernikos, and J. Parkhill, "Interpolated variable order motifs for identification of horizontally acquired DNA: revisiting the Salmonella pathogenicity islands," *Bioinformatics,* vol. 22, no. 18, pp. 2196-203, Sep 15, 2006.

[2] I. Rajan, S. Aravamuthan, and S. S. Mande, "Identification of compositionally distinct regions in genomes using the centroid method," *Bioinformatics,* vol. 23, no. 20, pp. 2672-7, Oct 15, 2007.

[3] S. Waack, O. Keller, R. Asper *et al.*, "Score-based prediction of genomic islands in prokaryotic genomes using hidden Markov models," *BMC Bioinformatics,* vol. 7, pp. 142, 2006.

[4] W. Hsiao, I. Wan, S. J. Jones *et al.*, "IslandPath: aiding detection of genomic islands in prokaryotes," *Bioinformatics,* vol. 19, no. 3, pp. 418-20, Feb 12, 2003.

[5] S. Shrivastava, V. Reddy Ch, and S. S. Mande, "INDeGenIUS, a new method for high-throughput identification of specialized functional islands in completely sequenced organisms," *J Biosci,* vol. 35, no. 3, pp. 351-64, Sep, 2010.

[6] Q. Tu, and D. Ding, "Detecting pathogenicity islands and anomalous gene clusters by iterative discriminant analysis," *FEMS Microbiol Lett,* vol. 221, no. 2, pp. 269-75, Apr 25, 2003.

[7] M. G. Langille, and F. S. Brinkman, "IslandViewer: an integrated interface for computational identification and visualization of genomic islands," *Bioinformatics,* vol. 25, no. 5, pp. 664-5, Mar 1, 2009.

[8] M. G. Langille, W. W. Hsiao, and F. S. Brinkman, "Evaluation of genomic island predictors using a comparative genomics approach," *BMC Bioinformatics,* vol. 9, pp. 329, 2008.

[9] J. Hu, B. Li, and D. Kihara, "Limitations and potentials of current motif discovery algorithms," *Nucleic Acids Res,* vol. 33, no. 15, pp. 4899-913, 2005.

[10] D. Che, S. Jensen, L. Cai *et al.*, "BEST: binding-site estimation suite of tools," *Bioinformatics,* vol. 21, no. 12, pp. 2909-11, Jun 15, 2005.

[11] E. Wijaya, S. M. Yiu, N. T. Son *et al.*, "MotifVoter: a novel ensemble method for fine-grained integration of generic motif finders," *Bioinformatics,* vol. 24, no. 20, pp. 2288-95, Oct 15, 2008.