THE UNIVERSITY OF CHICAGO

IDENTIFYING GENOMIC ISLANDS WITH DEEP NEURAL NETWORKS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE THESIS DIVISION
IN CANDIDACY FOR THE DEGREE OF
TYPE OF DEGREE

DEPARTMENT OF THESIS DEPARTMENT

BY

THESIS AUTHOR

CHICAGO, ILLINOIS
GRADUATION DATE
# TABLE OF CONTENTS

LIST OF FIGURES ......................................................... iv

LIST OF TABLES ......................................................... v

ACKNOWLEDGMENTS ..................................................... vi

ABSTRACT ................................................................. vii

1 GENOMIC ISLANDS ..................................................... 1
   1.1 Introduction ..................................................... 1
      1.1.1 Background and motivation ................................. 1
      1.1.2 Related Work ............................................... 2
      1.1.3 Challenges .................................................. 4
   1.2 Methods .......................................................... 5
      1.2.1 From Genes to Images ...................................... 5
      1.2.2 Why Images? ............................................... 6
      1.2.3 Deep Transfer Learning .................................... 9
   1.3 Validation ....................................................... 10
   1.4 Results .......................................................... 11

REFERENCES .......................................................... 17
## LIST OF FIGURES

1.1 Examples of images generated from the compare region viewer. Each directed arrow represents a gene color coded to match its functionality. The first row is the genome neighborhood of the focus gene (red), and the subsequent rows represent anchored regions from similar genomes sorted by their phylogenetic distances to the query genome. .............................................................. 7

1.2 ROC curve for our genomic island binary classifier. The ROC curve plots the true positive rate as a function of the false positive rate. The greater the area under the curve is (the closer it is to the unrealistic ideal top left corner point), the better. .................................................................................... 10

1.3 Examples of regions uniquely predicted by ShutterIsland .......................... 15

1.4 Alien Hunter predictions’ composition. Top 10 genes with the percentage of unique Alien Hunter predictions they reside in. ............................... 16

1.5 Shutter Island predictions’ composition. Top 10 genes with the percentage of unique Shutter Island predictions they reside in. ............................. 16
LIST OF TABLES

1.1 Number of islands and their total base pair value predicted by each tool over the testing genomes dataset. ................................. 13
1.2 Cross-tool comparison of GI results: % predicted GIs also predicted by other tools, over the testing genomes dataset. ............................. 13
1.3 Quality of tools predictions: % predictions made with GI features — % predictions missed with GI features, over the testing genomes dataset. ............ 14
1.4 Unique predictions made by each tool over the testing genomes dataset. ......... 14
ABSTRACT

Horizontal gene transfer is the main source of adaptability for bacteria, through which genes are obtained from different sources including bacteria, archaea, viruses, and eukaryotes. This process promotes the rapid spread of genetic information across lineages, typically in the form of clusters of genes referred to as genomic islands (GIs). Different types of GIs exist, often classified by the content of their cargo genes or their means of integration and mobility. Various computational methods have been devised to detect different types of GIs, but no single method currently is capable of detecting all GIs. We propose a method, which we call Shutter Island, that uses a deep learning model (Inception V3, widely used in computer vision) to detect genomic islands. The intrinsic value of deep learning methods lies in their ability to generalize. Via a technique called transfer learning, the model is pre-trained on a large generic dataset and then re-trained on images that we generate to represent genomic fragments. We demonstrate that this image-based approach generalizes better than the existing tools.
CHAPTER 1
GENOMIC ISLANDS

1.1 Introduction

1.1.1 Background and motivation

Horizontal gene transfer is the main source of adaptability for bacteria, through which genes are obtained from different sources including bacteria, archaea, viruses, and eukaryotes. This process promotes the rapid spread of genetic information across lineages, typically in the form of clusters of genes referred to as genomic islands (GIs). Different types of GIs exist, often classified by the content of their cargo genes or their means of integration and mobility. Various computational methods have been devised to detect different types of GIs, but no single method currently is capable of detecting all GIs.

Interest in genomic islands resurfaced in the 1990s, when some Escherichia coli strains were found to have exclusive virulence genes that were not found in other strains [1, 2]. These genes were thought to have been acquired by these strains horizontally and were referred to as pathogenicity islands (PAIs). Further investigations showed that other types of islands carrying other types of genes exist, giving rise to more names such as "secretion islands," "resistance islands," and "metabolic islands," since the genes carried by these islands could promote not only virulence but also symbiosis or catabolic pathways [3, 4, 5]. Aside from functionality, different names are also assigned to islands on the basis of their mobility. Some GIs are mobile and can thus move themselves to new hosts, such as conjugative transposons, integrative and conjugative elements (ICEs), and prophages, whereas other GIs lose their mobility [6, 7]. Prophages are viruses that infect bacteria and then remain inside the cell and replicate with the genome [8]. They are also referred to as bacteriophages in some literature, constituting the majority of viruses, and outnumbering bacteria by a factor of 10 to 1 [9, 10].

A genomic island (GI) then is a cluster of genes that is typically between 10 kb and 200 kb.
Horizontal gene transfer (HGT) may contribute to anywhere between 1.6% and 32.6% of the genomes [12, 13, 14, 15, 16, 17, 18, 19, 20]. This percentage implies that a major factor in the variability across bacterial species and clades can be attributed to GIs [21]. Thus, they impose an additional challenge to our ability to reconstruct the evolutionary tree of life. The identification of GIs is also important for the advancement of medicine, by helping develop new vaccines and antibiotics [22, 23] or cancer therapies [24]. For example, knowing that PAIs can carry many pathogenicity genes and virulence genes [25, 26, 27], researchers found that potential vaccine candidates resided within PAIs [28].

1.1.2 Related Work

While early methods focused on manual inspection of disrupted genomic regions that may resemble GI attachment sites [29] or show unusual nucleotide content [30, 31], the most recent computational methods fall into two broad categories: methods that involve sequence composition and methods that involve comparative genomics [1]. They both focus on one or more of the features that make GIs distinct, such as compositional bias, mobility elements, and transfer RNA (tRNA) hotspots [5, 7, 11, 25, 26, 32, 33, 34]. We discuss some of these features in more detail; they are listed by decreasing order of importance in [1, 35].

- One of the most important features of GIs is that they are sporadically distributed; that is, they are found only in certain isolates from a given strain or species.

- Since GIs are transferred horizontally across lineages and since different bacterial lineages have different sequence compositions, measures such as GC content or, more generally, oligonucleotides of various lengths (usually 2–9 nucleotides) are being used [36, 37, 38, 39, 40, 41, 42]. Codon usage is a well-known metric, which is the special case of oligonucleotides of length 3.

- Since the probability of having outlying measurements decreases as the size of the
region increases, tools usually use cut-off values for the minimum size of a region (or
gene cluster) to be identified as a GI.

- Another type of evidence comes not from the attachment sites but from what is in
  between, since some genes (e.g., integrases, transposases, phage genes) are known to
  be associated with GIs [25].

- In addition to the size of the cluster, evidence from mycobacterial phages [43] suggests
  that the size of the genes themselves is shorter in GIs than in the rest of the bacterial
  genome. Different theories suggest that this may confer mobility or packaging or
  replication advantages [8].

- Some GIs integrate specifically into genomic sites such as tRNA genes, introducing
  flanking direct repeats. Thus, the presence of such sites and repeats may be used as
  evidence for the presence of GIs [45, 46, 47].

Other research suggests that the directionality of the transcriptional strand and the protein
length are key features in GI prediction [8]. The available tools focus on one or more of the
mentioned features.

*Islander* works by first identifying tRNA and transfer-messenger RNA genes and their
fragments as endpoints to candidate regions, then disqualifying candidates through a set of
filters such as sequence length and the absence of an integrase gene [3]. *IslandPick* identifies
GIs by comparing the query genome with a set of related genomes selected by an evolutionary
distance function [44]. It uses Blast and Mauve for the genome alignment. The outcome
heavily depends on the choice of reference genomes selected. *Phaster* uses BLAST against
a phage-specific sequence database (the NCBI phage database and the database developed
by Srividhya et al. [48]), followed by DBSCAN [49] to cluster the hits into prophage regions.
*IslandPath-DIMOB* considers a genomic fragment to be an island if it contains at least one
mobility gene, in addition to 8 or more consecutive open reading frames with di-nucleotide
bias [50]. *SIGI-HMM* uses the Viterbi algorithm to analyze each gene’s most probable
codon usage states, comparing it against codon tables representing microbial donors or highly expressed genes, and classifying it as native or non-native accordingly [51]. PAI-IDA uses the sequence composition features, namely, GC content, codon usage, and dinucleotide frequency, to detect GIs [52]. Alien Hunter (or IVOM) uses k-mers of variable length to perform its analysis, assigning more weight to longer k-mers [38]. Phispy uses random forests to classify windows based on features that include transcription strand directionality, customized AT and GC skew, protein length, and abundance of phage words [8]. Phage Finder classifies 10 kb windows with more than 3 bacteriophage-related proteins as GIs [53]. IslandViewer is an ensemble method that combines the results of three other tools—SIGI-HMM, IslandPath-DIMOB, and IslandPick—into one web resource [54].

1.1.3 Challenges

No single tool is able to detect all GIs in all bacterial genomes [31]. Methods that narrow their search to GIs that integrate under certain conditions, such as into tRNAs, miss out on the other GIs. Similarly, not all GI regions exhibit atypical nucleotide content [22, 55]. Evolutionary events such as gene loss and genomic rearrangement [5] present more challenges. Also, highly expressed genes (e.g., genes in ribosomal protein operons), or those having an island host and donor that belong to the same or closely related species, or the fact that amelioration would pressure even genes from distantly related genomes to adapt to the host over time would lead to the host and the island exhibiting similar nucleotide composition [56] and subsequently to false negatives [23]. Tools that use windows face difficulty in adjusting their size: small sizes lead to large statistical fluctuation, whereas larger sizes result in low resolution [57].

For comparative genomics methods, the outcomes depend strongly on the choice of genomes used in the alignment process. Very distant genomes may lead to false positives, and very close genomes may lead to false negatives. In general, the number of reported GIs may differ across tools, because one large GI is often reported as a few smaller ones or vice
versa, also making it harder to detect end points and boundaries accurately, even with the use of hidden Markov models by tools such as AlienHunter and SIGI-HMM.

Furthermore, no reliable GI dataset exists that can validate the predictions of all these computational methods [38]. Although several databases exist, they usually cover only specific types of GIs [Islander, PAIDB, ICEberg], which would flag as false positives any extra predictions made by those tools. Moreover, as Nelson et al. state, “The reliability of the databases has not been verified by any convincing biological evidence” [6].

1.2 Methods

1.2.1 From Genes to Images

PATRIC (the Pathosystems Resource Integration Center) is a bacterial bioinformatics resource center that we are part of (https://www.patricbrc.org)[58]. It provides researchers with the tools necessary to analyze their private data and to compare it with public data. PATRIC recently surpassed the 200,000 publicly sequenced genomes mark, ensuring that enough genomes are available for effective comparative genomics studies. PATRIC provides a compare region viewer service, where a query genome is aligned against a set of other related genomes anchored at a specific focus gene. The service finds other genes that are of the same family as the query gene and then aligns their flanking regions accordingly. Such graphical representations are appealing because they help users visualize the genomic areas of interest. In the resulting plots, genomic islands should appear as gaps in alignment as opposed to conserved regions. We replicated the service by implementing an offline version forked from the production user interface, which is necessary for computational efficiency and for consistency in the face of any future interface changes.

Figure 2.1 shows sample visualizations of different genomic fragments belonging to the two classes. Each row represents a region in a genome, with the query genome being the top row. Each arrow represents a single gene, capturing its size and strand directionality. Colors
represent functionality. The red arrow represents the query gene, at which the alignment with the rest of the genomes is anchored. The remaining genes share the same color if they belong to the same family; they are colored black if they are not found in the query genome’s focus region. Some colors are reserved for key genes: green for mobility genes, yellow for tRNA genes, and blue for phage related genes.

Figures 2.1a and 2.1b are examples of a query genome with a non-conserved neighborhood. The focus peg lacks alignments in general or is being aligned with genes from other genomes with different neighborhoods, containing genes with different functionalities from those in the query genome (functionality is color coded). In contrast, Figures 2.1c and 2.1d show more conserved regions, which are what we expect to see in the absence of GIs.

1.2.2 Why Images?

Representing genomic fragments as images makes it easier to leverage the powerful machine learning (ML) technologies that have become the state of art in solving computer vision problems. Algorithms based on deep neural networks have proven to be superior in competitions such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRS) [59]. Deep learning is the process of training neural networks with many hidden layers. The depth of these networks allows them to learn more complex patterns and higher-order relationships, at the cost of being more computationally expensive and requiring more data to work effectively. Improvements in such algorithms have been translated to improvements in a variety of domains reliant on computer vision tasks [60]. Our initial inspiration for representing genome features as images came from observing how human annotators work. These experts often examine the “compare region” view for a long time before they decide on the gene identity. A critical piece of information they rely on is how the focus gene compares with its homologs in related genomes. This information is cumbersome to represent in tabular data because (1) explicit all-to-all comparison is computationally expensive; (2) the comparisons need to be done at both individual gene and cluster levels including coordinates,
Figure 1.1: Examples of images generated from the compare region viewer. Each directed arrow represents a gene color coded to match its functionality. The first row is the genome neighborhood of the focus gene (red), and the subsequent rows represent anchored regions from similar genomes sorted by their phylogenetic distances to the query genome.

length, and neighborhood similarities; and (3) human experts integrate all these different levels of information with an intuition for fuzzy comparison, something that is hard to repli-
cate in tabular learning without additional parameterization or augmentation. Representing genomic features as images mitigates all three issues.

Images offer a natural way to compare genes (horizontally) and clusters across genomes (vertically) with 2D convolution. The fact that the compare region view sorts genomes by evolutionary distance allows the neural network to exploit locality and place more emphasis on close genomes via incremental pooling. An additional benefit of working with images is to be able to leverage the state-of-the-art deep learning models, many of which were first developed in vision tasks and perfected over years of iterations. Google researchers have used spectrograms (instead of text) in direct speech translation [61] and DNA sequence pile-up graphs (instead of alignment data) in genetic variant calling [62]. In both cases, the image-based models outperformed their respective previous state-of-the-art method based on traditional domain features. Further, the low-level visual feature patterns learned in pretrained image models have been demonstrated to transfer to distant learning tasks on non-image data in several preliminary studies ranging from environmental sound classification to cancer gene expression typing [63]. Much like feature-engineering methods, casting tabular data to images encodes information in a way more amenable to learning without explicitly adding information. It can also be easily integrated with other data modalities in the latent vector representation to prevent information loss. We hypothesize that this emerging trend of representing data with images will continue until model tuning and large-scale pretraining in scientific domains start to catch up with those in computer vision.

The images generated by PATRIC capture many of the most important GI features mentioned earlier—the sporadic distribution of islands, the protein length, functionality, and strand directionality—using color-coded arrows of various sizes. So, while PATRIC provides a lot of genomic data, the challenge comes down to building a meaningful training dataset. The databases available are still limited in size and specific in content, which in turn limits the ability even for advanced and deep models to learn and generalize well.
1.2.3 Deep Transfer Learning

Training deep models over a limited dataset puts the model at the risk of overfitting. One way around this problem is using a technique referred to as transfer learning[64]. In transfer learning, a model does not have to be trained from scratch. Instead, the idea is to retrain a model that has been previously trained on a related task. The newly retrained model should then be able to transfer its existing knowledge and apply it to the new task. This approach affords the ability to reuse models that have been trained on huge amounts of data, while adding the necessary adjustments to make them available to work with more limited datasets, adding a further advantage to our approach of representing the data visually.

In our approach (Shutter Island), we use Google’s Inception V3 architecture that has been previously trained on ImageNet. The Inception V3 architecture is a 48-layer-deep convolutional neural network [60]. Training such a deep network on a limited dataset such as the one available for GIs is unlikely to produce good results. ImageNet is a database that contains more than a million images belonging to more than a thousands categories [59]. The ImageNet project runs the ILSVRC dataset annually. The Inception V3 model reaches a 3.5% top-5 error rate on the 2012 ILSVRC dataset, where the winning model that year had a 15.3% error rate. Thus, a model that was previously trained on ImageNet is already good at feature extraction and visual recognition. To make the model compatible with the new task, the top layer of the network is retrained on our GI dataset, while the rest of the network is left intact, a strategy that is more powerful than starting with a deep network with random weights.

For our training data, we used the set of reference+representative genomes found on PATRIC. For each genome, our program produced an image for every non-overlapping 10 kbp window. A balanced dataset was then curated from the total set of images created. Since this is a supervised learning approach and our goal is to generalize over the tools’ predictions and beyond, we used Phispy and IslandViewer’s predictions to label the images that belong to candidate islands. IslandViewer captures the predictions of different methods
that follow different approaches, and Phispy captures different GI features. Although the primary goal is to predict the union of the predictions of other tools and to generalize, we labeled a genomic fragment as a GI only if it belonged to the intersection of the predictions made by these tools. Doing so increases confidence that a certain candidate island is actually so. Overall, the intersection of these tools’ predictions spanned only about half of the genome dataset. Our model reached a training accuracy of 92%, with a validation accuracy of 88%.

Figure 2.2 shows the receiver operating characteristic (ROC) curve of our classifier. We considered a predicted gene to be a true positive if it overlaps with another tool’s prediction or falls within a region that shows GI features, and a false positive otherwise. We considered a gene to be a true negative if it does not overlap with any tool’s predictions or does not fall in an area that includes GI features, and a false negative otherwise. For the areas of interest, we used windows of four surrounding genes on each side.

Figure 1.2: ROC curve for our genomic island binary classifier. The ROC curve plots the true positive rate as a function of the false positive rate. The greater the area under the curve is (the closer it is to the unrealistic ideal top left corner point), the better.

1.3 Validation

Since no reliable benchmark is available, we used the set of genomes mentioned in Phispy [8]. The set consists of 41 bacterial genomes that include 190 GIs. This set served as a good
common ground for all the tools we mentioned. Some of those tools have not been updated for a while, but all the tools had predictions made over the genomes in this set. The authors of Phispy reported that the GIs have been manually verified [8]. We discarded the genomes that caused errors with any of the tools used in the comparison and any genomes that were part of the training set. All the presented results are aggregates over the mentioned genome dataset. When treating novel genomes, the same compare genome viewer service was used, aligning the query genome with the set of reference+representative genomes. The only difference is that an image is created for every gene in the genome, providing better resolution than with non-overlapping windows. Next, each window is classified as either part of a GI or not. That label belongs to the focus gene in that window. Eventually, every gene in the genome has a label, and these are clustered into GIs with a minimum cut-off value of 8 kbp.

In addition to using powerful technologies and extensive data, this approach may add an extra advantage over whole-genome alignment methods because performing the alignment over each gene may provide a higher local resolution and aid in resisting evolutionary effects such as recombination and others that may have happened after the integration and that usually affect GI detection efforts.

1.4 Results

We used a dataset containing manually verified GIs as our testing set. Table 2.1 shows a summary of every tool’s predictions over the entire genome dataset, listing the number of islands and their total base pair count. Different tools follow different custom-defined metrics to judge their results, typically by using a threshold representing the minimum values of features (e.g., number of phage words) present in a region to be considered a GI. Admittedly, judging the results is not a trivial task; otherwise the problem of detecting GIs would have been largely solved by the validation method itself. Since one of the main challenges is the absence of a reliable benchmark and a trivial way to verify the tools’
predictions, we judge the results based on a combination of three factors reflecting on
the overlap and uniqueness of tool predictions and how well they capture gene annotation. First,
we consider the percentage of each tool’s results that every tool predicts (Table 2.2). This
gives us a measure of how much a single tool can generalize and how much of the union
of all tools’ predictions it can get. Second, since some tools make many more predictions
than do others, to get a better idea about the quality of the predictions made and missed
by the other tools, we show in Table 2.3 a breakdown of the percentage of islands with
known GI features (e.g., tRNA, mobility, phage genes). Tools that use these features to
perform their classifications were omitted. To identify these GI features, we scoured the
literature and identified certain keywords found in gene annotations that are related to GIs.
Such annotations of gene identity are either directly curated by humans or reflect human
assessment through exemplar-based computational propagation. We constructed a standard
vocabulary of the GI-related keywords that were also in agreement with the more extensive
list of keywords used by Phaster [21] for the same purpose.

Third, we consider each tool’s unique predictions and their quality based on the presence
of GI features in their content (Table 2.4).

Another challenge in assessing GI prediction is getting precise endpoints for predicted
islands. Since different tools report a different number of islands owing to the nature of the
features they use, where one island could be reported as many or vice versa, we considered
a tool to predict another’s islands if any of its predictions overlap with the other tool’s
predictions. We counted the percentage of base pair coverage of that other tool as represented
by its predicted endpoints. This allowed us to compare overlapping islands predicted by
different tools even if their coordinates did not match. In Table 2.2, we include the tools
that we were able to run. Default parameters were used for all tools.

From Table 2.1 we can see that AlienHunter calls the most GIs, with almost double the
amount called by ShutterIsland and IslandViewer if measured by base pair count, and even
more if measured by island count. ShutterIsland’s prediction count is close to IslandViewer’s,
Table 1.1: Number of islands and their total base pair value predicted by each tool over the testing genomes dataset.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Number of Islands</th>
<th>Number of Base Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alien Hunter</td>
<td>1919</td>
<td>19,561,593</td>
</tr>
<tr>
<td>ShutterIsland</td>
<td>649</td>
<td>10,700,492</td>
</tr>
<tr>
<td>IslandViewer</td>
<td>701</td>
<td>10,571,974</td>
</tr>
<tr>
<td>IslandPath-Dimob</td>
<td>331</td>
<td>6,871,312</td>
</tr>
<tr>
<td>Phaster</td>
<td>109</td>
<td>4,334,225</td>
</tr>
<tr>
<td>Phispy</td>
<td>96</td>
<td>3,979,173</td>
</tr>
<tr>
<td>Phage Finder</td>
<td>85</td>
<td>3,656,950</td>
</tr>
<tr>
<td>IslandPick</td>
<td>356</td>
<td>3,020,733</td>
</tr>
<tr>
<td>SIGI-HMM</td>
<td>329</td>
<td>2,543,145</td>
</tr>
<tr>
<td>Islander</td>
<td>50</td>
<td>2,019,610</td>
</tr>
</tbody>
</table>

which is remarkable considering that IslandViewer is an ensemble method combining four other tools’ predictions.

Table 1.2: Cross-tool comparison of GI results: % predicted GIs also predicted by other tools, over the testing genomes dataset.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>ShutterIsland</th>
<th>IslandViewer</th>
<th>Phispy</th>
<th>PhageFinder</th>
<th>Islander</th>
<th>Phaster</th>
<th>Alien Hunter</th>
<th>IslandPick</th>
<th>Dimob</th>
<th>SIGI</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShutterIsland</td>
<td>N/A</td>
<td>45.7%</td>
<td>97.8%</td>
<td>99.1%</td>
<td>67.4%</td>
<td>92.9%</td>
<td>27%</td>
<td>20.3%</td>
<td>54%</td>
<td>28.8%</td>
</tr>
<tr>
<td>IslandViewer</td>
<td>42.8%</td>
<td>N/A</td>
<td>89.3%</td>
<td>89.2%</td>
<td>N/A</td>
<td>82.1%</td>
<td>39.4%</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Phispy</td>
<td>29.1%</td>
<td>23.7%</td>
<td>N/A</td>
<td>98.3%</td>
<td>52.8%</td>
<td>79.3%</td>
<td>9%</td>
<td>10.8%</td>
<td>29.1%</td>
<td>11.5%</td>
</tr>
<tr>
<td>PhageFinder</td>
<td>28.1%</td>
<td>23.6%</td>
<td>92.8%</td>
<td>N/A</td>
<td>50.4%</td>
<td>79.8%</td>
<td>9%</td>
<td>10.1%</td>
<td>29/3%</td>
<td>12%</td>
</tr>
<tr>
<td>Islander</td>
<td>9.2%</td>
<td>21.2%</td>
<td>23.7%</td>
<td>25.7%</td>
<td>N/A</td>
<td>22.2%</td>
<td>8.3%</td>
<td>15.5%</td>
<td>22.9%</td>
<td>17%</td>
</tr>
<tr>
<td>Phaster</td>
<td>26.4%</td>
<td>22.5%</td>
<td>82.4%</td>
<td>86%</td>
<td>44.5%</td>
<td>N/A</td>
<td>10.4%</td>
<td>11.3%</td>
<td>27.5%</td>
<td>12.7%</td>
</tr>
<tr>
<td>AlienHunter</td>
<td>56.8%</td>
<td>78.9%</td>
<td>87.2%</td>
<td>86.5%</td>
<td>98%</td>
<td>87.2%</td>
<td>N/A</td>
<td>67.1%</td>
<td>82.8%</td>
<td>92.6%</td>
</tr>
<tr>
<td>IslandPick</td>
<td>10.6%</td>
<td>43.3%</td>
<td>25.4%</td>
<td>28.7%</td>
<td>51.7%</td>
<td>29%</td>
<td>13.8%</td>
<td>N/A</td>
<td>28.4%</td>
<td>31.2%</td>
</tr>
<tr>
<td>Dimob</td>
<td>34.9%</td>
<td>70.5%</td>
<td>86.1%</td>
<td>85.2%</td>
<td>87.8%</td>
<td>76.9%</td>
<td>25.7%</td>
<td>29.5%</td>
<td>N/A</td>
<td>50.3%</td>
</tr>
<tr>
<td>SIGI</td>
<td>17.2%</td>
<td>47.7%</td>
<td>34.4%</td>
<td>31.6%</td>
<td>63.2%</td>
<td>27.8%</td>
<td>22.6%</td>
<td>30.9%</td>
<td>44.8%</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 2.2 shows that AlienHunter has the biggest coverage in general when it comes to predicting other tools’ results. This coverage is expected given that its predictions’ base pair coverage is almost 10 times as much as the tools with least coverage. ShutterIsland comes next and predicts the most out of three tools’ predictions. What is clear is the model’s ability to generalize, considering that it was trained only on the intersection of the predictions made.
by Phispy and IslandViewer but nevertheless also got the most predictions for other tools such as PhageFinder and Phaster. Moreover, we note that specific tools such as Islander detect only a subset of the results, whereas the rest of the tools score somewhere in between.

Table 1.3: Quality of tools predictions: % predictions made with GI features — % predictions missed with GI features, over the testing genomes dataset.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>ShutterIsland</th>
<th>IslandViewer</th>
<th>AlienHunter</th>
<th>IslandPick</th>
<th>SIGI</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShutterIsland</td>
<td>N/A</td>
<td>91% — 64%</td>
<td>87% — 47%</td>
<td>89% — 31%</td>
<td>87% — 36%</td>
<td><strong>89% — 45%</strong></td>
</tr>
<tr>
<td>IslandViewer</td>
<td>94% — 67%</td>
<td>N/A</td>
<td>89% — 45%</td>
<td>80% — n/a</td>
<td>87% — n/a</td>
<td><strong>88% — 56%</strong></td>
</tr>
<tr>
<td>AlienHunter</td>
<td>74% — 70%</td>
<td>66% — 60%</td>
<td>N/A</td>
<td>73% — 21%</td>
<td>71% — 42%</td>
<td><strong>71% — 48%</strong></td>
</tr>
<tr>
<td>IslandPick</td>
<td>69% — 76%</td>
<td>34% — 86%</td>
<td>49% — 53%</td>
<td>N/A</td>
<td>54% — 44%</td>
<td><strong>52% — 65%</strong></td>
</tr>
<tr>
<td>SIGI</td>
<td>67% — 75%</td>
<td>45% — 77%</td>
<td>48% — 51%</td>
<td>50% — 35%</td>
<td>N/A</td>
<td><strong>53% — 60%</strong></td>
</tr>
<tr>
<td>Dimob</td>
<td>n/a — 66%</td>
<td>n/a — 28%</td>
<td>n/a — 43%</td>
<td>n/a — 25%</td>
<td>n/a — 23%</td>
<td>n/a — 37%</td>
</tr>
<tr>
<td>Phispy</td>
<td>n/a — 68%</td>
<td>n/a — 70%</td>
<td>n/a — 50%</td>
<td>n/a — 33%</td>
<td>n/a — 39%</td>
<td>n/a — 52%</td>
</tr>
<tr>
<td>PhageFinder</td>
<td>n/a — 68%</td>
<td>n/a — 70%</td>
<td>n/a — 50%</td>
<td>n/a — 34%</td>
<td>n/a — 39%</td>
<td>n/a — 52%</td>
</tr>
<tr>
<td>Islander</td>
<td>n/a — 75%</td>
<td>n/a — 71%</td>
<td>n/a — 51%</td>
<td>n/a — 33%</td>
<td>n/a — 39%</td>
<td>n/a — 54%</td>
</tr>
<tr>
<td>Phaster</td>
<td>n/a — 75%</td>
<td>n/a — 71%</td>
<td>n/a — 51%</td>
<td>n/a — 33%</td>
<td>n/a — 39%</td>
<td>n/a — 54%</td>
</tr>
</tbody>
</table>

From Table 2.3, we can see that on average ShutterIsland is the tool with most predictions showing GI features being missed by other tools. It is also the tool that calls the most predictions showing GI features and misses the fewest such predictions made by other tools. So even though AlienHunter makes more predictions in general, more predictions made by ShutterIsland exhibit known GI features.

Table 1.4: Unique predictions made by each tool over the testing genomes dataset.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Unique (count)</th>
<th>Unique (bp)</th>
<th>with GI features (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alien Hunter</td>
<td>1155</td>
<td>9,583,497</td>
<td>40%</td>
</tr>
<tr>
<td>Shutter Island</td>
<td>280</td>
<td>3,647,377</td>
<td>65%</td>
</tr>
<tr>
<td>Phaster</td>
<td>2</td>
<td>30,814</td>
<td>0%</td>
</tr>
<tr>
<td>Phispy</td>
<td>1</td>
<td>26,890</td>
<td>100%</td>
</tr>
</tbody>
</table>

Both ShutterIsland and AlienHunter have a lot of unique predictions, as is clear in Table 2.4. AlienHunter’s unique predictions alone are almost more than every other tool’s total predictions. They average 8 kbp in length. ShutterIsland’s unique predictions are also more than most of the other tools’ predictions, with an average length of 14 kbp. Applying the same length cutoff threshold (8 kbp) on AlienHunter’s unique predictions reduces them
to 301 islands with a total of 3,880,000 bp, which is on par with ShutterIsland’s unique predictions. Unlike AlienHunter, however, most of ShutterIsland’s unique predictions show GI features. In Figure 2.3 we present some snapshots of typical unique predictions made by ShutterIsland, in addition to a breakdown of the most frequent gene annotations that are included in those predictions (Figures 2.4 and 2.5).
Figure 1.4: Alien Hunter predictions’ composition. Top 10 genes with the percentage of unique Alien Hunter predictions they reside in.

- hypothetical protein
- Mobile element protein
- putative membrane protein
- Transposase
- putative lipoprotein
- Transcriptional regulator LysR family
- DNA-directed RNA polymerase beta subunit
- Transaldolase
- Rhs-family protein
- Methyl-accepting chemotaxis protein
- Translation elongation factor G
- DNA-directed RNA polymerase beta
- SSU ribosomal protein S12p (S23e)
- SSU ribosomal protein S7p (S5e)

Figure 1.5: Shutter Island predictions’ composition. Top 10 genes with the percentage of unique Shutter Island predictions they reside in.

- hypothetical protein
- Mobile element protein
- putative membrane protein
- putative lipoprotein
- ABC transporter ATP-binding protein
- Putative periplasmic protein
- Transcriptional regulator
- Transposase
- Putative inner membrane protein
- PilS cassette
- putative integral membrane protein
REFERENCES


[57] Zhang, R., Zhang, C.T. (2004) A systematic method to identify genomic islands and
its applications in analyzing the genomes of Corynebacterium glutamicum and Vibrio

Disz, T., Gabbard, J.L., Gerdes, S., Henry, C.S., Kenyon, R.W., Machi, D., Mao, C.,
Nordberg, E.K., Olsen, G.J., Murphy-Olson, D.E., Olson, R., Overbeek, R., Parrello,
B., Pusch, G.D., Shukla, M., Vonstein, V., Warren, A., Xia, F., Yoo, H., Stevens, R.L.
(2017) Improvements to PATRIC, the all-bacterial Bioinformatics Database and Analysis

[59] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy,
recognition challenge. *IJCV*.

Inception Architecture for Computer Vision. *The IEEE Conference on Computer Vision
and Pattern Recognition (CVPR)*, 2818–2826.

Direct speech-to-speech translation with a sequence-to-sequence model. arXiv preprint

Biotechnology*

from scratch. Retrieved from https://www.youtube.com/watch?v=ccMHJeQU4Qw