Bioinformatic analysis of RNA from diverse species

Ambarish Biswas

a thesis submitted for the degree of
Doctor of Philosophy
at the University of Otago, Dunedin,
New Zealand.
10 December 2013
Abstract

Although most living cells contain a full complement of genes, gene expression differs between cells. In addition to being translated, RNA molecules play key roles in other important cellular functions. Gene expression in eukaryotic cells is controlled post-transcriptionally by combinations of regulatory elements in mRNAs. Such elements include structural and functional sequence motifs, and binding sites for miRNAs and proteins. These are usually located in the untranslated regions (UTRs) of mRNA sequences, particularly the 3’UTRs. Characterisation of these elements can provide detailed insight into the complex mechanism of regulation of gene expression. However, identification and visualisation of these cis-regulatory elements in a true overlapping manner is a challenging task. The Scan for Motifs (SFM) web-application simplified the process of identifying a wide range of regulatory elements on alignments of vertebrate 3’UTRs as well as in any inputted sequence. SFM includes identification of both RNA Binding Protein (RBP) sites and targets of miRNAs. The regulatory elements can be filtered by False Discovery Rate (FDR) estimations. The output provides an interactive graphical representation highlighting potential regulatory elements and overlaps between them complemented with simple statistics and cross-reference to their sources.

In eukaryotes, the RNAi (RNA interference) is now a well-studied and established method for inhibiting gene expression. A similar system, known as CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) or Cas (CRISPR associated) system is present in bacteria and archaea that provides adaptive immunity against phage infection. CRISPR RNAs (crRNAs) are a type of small noncoding RNA that plays an important role in a noncoding RNA guided defence system in prokaryotes. Specific prediction methods found crRNA-encoding loci in nearly half of sequenced bacterial, and ~90% of archaeal species. The bacterial and archaeal CRISPR/Cas adaptive immune system targets specific protospacer nucleotide sequences in invading organisms. This requires base pairing between processed CRISPR RNA and the target protospacer. For type I and II CRISPR/Cas systems, protospacer adjacent motifs (PAM) are essential for target recognition, and for type III, mismatches in the flanking sequences are important in the antiviral response. In this study, the properties of each class of CRISPR were examined and these information is used for building a tool.
CRISPRTarget) that predicts the most likely targets of CRISPR RNAs. This can be used to discover targets in newly sequenced genomic or metagenomic data. To test its utility, the features and targets of well-characterized *Streptococcus thermophilus* and *Sulfolobus solfataricus* type II and III CRISPR/Cas systems were discovered. Finally, in *Pectobacterium* species, new CRISPR targets were identified, establishing a model of temperate phage exposure and subsequent inhibition by the type I CRISPR/Cas systems.

CRISPR arrays consist of repeat sequences separated by specific spacer sequences. Generally one strand is transcribed, producing long pre-crRNAs, which are processed to short crRNAs that base pair with invading nucleic acids to facilitate their destruction. No current software for the discovery of CRISPR loci predicts the direction of crRNA transcription. A new algorithm (CRISPRDirection) was developed that accurately predicts the strand of the resulting crRNAs. The method supports FASTA/multi-FASTA sequence or repeats as well as a complete annotation file as input. CRISPRDirection uses parameters that are calculated from the CRISPR repeat predictions and flanking sequences, which are combined by weighted voting. The prediction may utilise optional prior coding sequence annotation. CRISPRDirection correctly predicted the orientation of 94% of a reference set of arrays.

Existing CRISPR detection algorithms do not utilise recently identified features of CRISPR structure, expression, or direction of RNA transcription. A series of routines were developed and implemented as CRISPRDetect that detect and refine CRISPR arrays. This algorithm is optimised but parameters are user tuneable. CRISPRDetect discovers putative arrays, extends the array by detecting additional repeats, and refines the internal structure in a true array specific manner. It also includes the direction of transcription by calculating and using parameters relating to the structure and evolution of the arrays. CRISPRDetect enables more accurate detection of arrays and is suitable for inclusion in genome annotation pipelines. It comes with an interactive web-server as well as a command-line executable. Additionally, an interactive database named CRISPRBank was developed, which contains CRISPR specific information from all published bacterial and archaeal genomes.
All these tools and associated files can be accessed at the bioanalysis server\(^1\), which as yet has the most comprehensive and diverse set of tools to aid CRISPR analysis.

\(^1\) http://bioanalysis.otago.ac.nz
Acknowledgements

Completing my PhD would have been lot harder if I did not have Dr Chris Brown as my supervisor. Ever since I applied for this PhD, he has been both kind and cooperative. Not only he guided me though the scholarship process, but also helped me relocate to New Zealand. Before coming to New Zealand, I did not know anyone here. But that has never been a problem. Chris single-handed managed everything. I never knew anyone that kind or even expected it. Having a computer science background, it would have way more difficult, if Chris would have not spent those countless hours explaining even the smallest details of Biochemistry. His patience and vast knowledge always motivated me. I think, I could do another PhD with him.

Maintaining a family in a scholarship is always difficult, especially with a child. I'm so grateful to my family, for all the sacrifices they made. Specially, to my son, who probably never realised that weekends are not working days. I will never forget how sadly he asked me almost every Saturday and Sundays in the last year 'papa, you going to office?'. I hope someday when he is grown up to read this, he will realise that I missed him equally. It would be impossible for me to finish the PhD in time, if my wife was not there for me. Whether it is taking our son to ‘Kindi’, doctor or doing the groceries, I appreciate how she managed everything on her own without complaining. I'm equally lucky and blessed to have very supportive parents, who always sacrificed their needs to provide me a better life. Without them, I would not have reached this far.

Before coming to New Zealand, I had no idea how life would be here. But so far, it is excellent. New Zealand is probably the most beautiful country and people here even nicer. I never thought that I would have so many good friends here, who kept in constant touch, even though I was constantly denying for most social events. The acknowledgement will be incomplete without appreciating my lab mates and a very friendly department, which almost became my second home.
Publications associated with this thesis


- Biswas, A., Fineran, P.C. and Brown, C.M. (submitted), Computational detection of CRISPR/crRNA targets to be published in "CRISPR: Methods and Protocols" part of the series "Methods in Molecular Biology" Humana Press, USA.

# List of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARE</td>
<td>AU-rich element</td>
</tr>
<tr>
<td>AJAX</td>
<td>Asynchronous JavaScript and XML</td>
</tr>
<tr>
<td>BED</td>
<td>Browser Extensible Data</td>
</tr>
<tr>
<td>Cas</td>
<td>CRISPR associated</td>
</tr>
<tr>
<td>CDS</td>
<td>CoDing Sequences</td>
</tr>
<tr>
<td>CREs</td>
<td>Cis-Regulatory Elements</td>
</tr>
<tr>
<td>CRISPR</td>
<td>Clustered Regularly Interspaced Short Palindromic Repeat</td>
</tr>
<tr>
<td>crRNA</td>
<td>CRISPR RNA</td>
</tr>
<tr>
<td>crRNP</td>
<td>CRISPR Ribonucleo Protein</td>
</tr>
<tr>
<td>DRs</td>
<td>Direct Repeats</td>
</tr>
<tr>
<td>FDR</td>
<td>False Discovery Rate</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>IRE</td>
<td>Iron Responsive Element</td>
</tr>
<tr>
<td>IUPAC</td>
<td>International Union of Pure and Applied Chemistry</td>
</tr>
<tr>
<td>JCVI</td>
<td>J. Craig Venter Institute</td>
</tr>
<tr>
<td>LAMP</td>
<td>Linux Apache MySQL and Perl</td>
</tr>
<tr>
<td>MFE</td>
<td>Minimum Free Energy</td>
</tr>
<tr>
<td>MSA</td>
<td>Multiple Sequence Alignments</td>
</tr>
<tr>
<td>ncRNA</td>
<td>non-coding RNA</td>
</tr>
<tr>
<td>NGS</td>
<td>Next Generation Sequencing</td>
</tr>
<tr>
<td>PAM</td>
<td>Protospacer Adjacent Motifs</td>
</tr>
<tr>
<td>PDB</td>
<td>Protein Data Bank</td>
</tr>
<tr>
<td>PFM</td>
<td>Position Frequency Matrix</td>
</tr>
<tr>
<td>PSSM</td>
<td>Position Specific Probability Matrix</td>
</tr>
<tr>
<td>PWM</td>
<td>Position Weight Matrix</td>
</tr>
<tr>
<td>RBP</td>
<td>RNA Binding Protein</td>
</tr>
<tr>
<td>RBPDB</td>
<td>RNA Binding Protein DataBase</td>
</tr>
<tr>
<td>RBPs</td>
<td>RNA binding proteins</td>
</tr>
<tr>
<td>REs</td>
<td>Regulatory Elements</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>RNAi</td>
<td>RNA interference</td>
</tr>
<tr>
<td>SAM</td>
<td>Spacer Acquisition Motifs</td>
</tr>
<tr>
<td>SFM</td>
<td>Scan For Motifs</td>
</tr>
<tr>
<td>TFBS</td>
<td>TF Binding Sites</td>
</tr>
<tr>
<td>TIM</td>
<td>Target Interference Motifs</td>
</tr>
<tr>
<td>TNF</td>
<td>Tumor Necrosis Factor</td>
</tr>
<tr>
<td>tracrRNA</td>
<td>Tracer RNA</td>
</tr>
<tr>
<td>TRF</td>
<td>Tandem Repeats Finder</td>
</tr>
<tr>
<td>UCSC</td>
<td>University of California, Santa Cruz</td>
</tr>
<tr>
<td>UTRs</td>
<td>Untranslated Regions</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Table of contents

Chapter 1: Introduction ........................................................................................................ 19
  1.1 Analysis of regulatory elements in eukaryotic RNA .................................................... 24
  1.1.1 Handling input sequence data ........................................................................... 25
  1.1.2 Identification of regulatory elements in RNA ..................................................... 26
  1.1.3 Processing putative regulatory elements ............................................................. 28
  1.1.4 Mapping regulatory elements and genes ............................................................. 31
  1.1.5 Visualisation ...................................................................................................... 32
  1.2 Analysis of CRISPR/Cas systems in prokaryotic RNA ............................................. 33
  1.2.1 Elements associated with the CRISPR/Cas system .......................................... 33
  1.2.2 Mechanism of action of the CRISPR/Cas defence system ................................. 36
  1.2.3 Types of CRISPR/Cas system ......................................................................... 37
  1.2.4 CRISPR bioinformatics ..................................................................................... 39
  1.3 Aims of this work ..................................................................................................... 42

Chapter 2: Scan for motifs: a tool for the analysis of combinations of regulatory elements in vertebrate 3’ UTRs .............................................................. 43
  2.1 Introduction ............................................................................................................. 43
  2.2 Materials and methods .......................................................................................... 45
  2.2.1 Obtaining, cleaning and formatting the data ...................................................... 46
  2.2.2 Handling user input ........................................................................................... 47
  2.2.3 Processing input ................................................................................................ 49
  2.3 Result ...................................................................................................................... 50
  2.4 Discussion .............................................................................................................. 53

Chapter 3: CRISPRTarget: Bioinformatic prediction and analysis of crRNA targets .................................................................................................................. 55
  3.1 Abstract ................................................................................................................... 55
  3.2 Introduction ............................................................................................................. 56
  3.3 Materials and Methods .......................................................................................... 59
  3.4 Results ...................................................................................................................... 62
  3.5 Discussion .............................................................................................................. 77

Chapter 4: Accurate computational prediction of the transcribed strand of CRISPR noncoding RNAs ................................................................................................. 81
  4.1 Abstract ................................................................................................................... 81
  4.2 Introduction ............................................................................................................. 82
  4.3 Methods .................................................................................................................. 85
  4.3.1 Preprocessing of arrays ...................................................................................... 85
  4.3.2 Calculation of Parameters .................................................................................. 88
  4.4 Results ...................................................................................................................... 92
  4.4.1 Specific sequences at the 3’ end of the repeat ...................................................... 93
  4.4.2 Biased AT composition in the 5’ and 3’ flanks ......................................................... 93
  4.4.3 Degeneracy at the 3’ end of the array .................................................................. 94
  4.4.4 Potential for RNA secondary structure in the repeat ............................................. 95
  4.4.5 Analysis of A relative to T in repeats – A/T ratio analysis .................................... 97
  4.4.6 Distance to the next coding gene at the 5’ and 3’ end- length of leader analysis .................................................................................................................. 97
  4.4.7 Combined Prediction ......................................................................................... 99
  4.5 Discussion ..............................................................................................................100
4.5.1 Reference Set ........................................................................................................ 101
4.5.2 Specific methods .................................................................................................. 102

Chapter 5: CRISPRDetect: A flexible algorithm to define CRISPR arrays........107
5.1 Abstract .................................................................................................................. 107
5.2 Introduction ............................................................................................................ 108
5.3 Materials and Methods ......................................................................................... 110
5.3.1 Detection of direct repeats - putative CRISPRs ............................................. 111
5.3.2 Removal of tandem repeats ............................................................................ 112
5.3.3 Refinement of putative CRISPR ................................................................. 112
5.3.4 Finding direction and predicting subtype ....................................................... 118
5.3.5 Scoring the predictions .................................................................................... 118
5.4 Results and Discussion ......................................................................................... 118
5.4.1 Overview: ......................................................................................................... 118
5.4.2 Refinement of arrays internal structure ......................................................... 123
5.4.3 Supplementary analysis .................................................................................. 129
5.5 Conclusion ............................................................................................................. 130

Chapter 6: Conclusion ................................................................................................. 131

Future Directions ........................................................................................................ 133

References .................................................................................................................. 135

Supporting tables and contents for Chapter 2 ......................................................... 157
A.1.Table UTR_Sequences ......................................................................................... 158
A.2.Table Predicted_Targets_Info ........................................................................... 158

Supporting tables and contents for Chapter 4 ......................................................... 159
B.1. Reference set of repeats ...................................................................................... 160
B.2. Flow diagram of processes in CRISPRDirection.pl ........................................ 161
B.3. Repeat classes covered in the list of reference repeats ................................... 162
B.4. Combinations of analysis (leaving out each in turn) for 100 random arrays .. 163
B.5. Relative AT richness in the two flanks .............................................................. 165

Supporting tables and contents for Chapter 5 ......................................................... 167
C.1. Array scoring scheme ......................................................................................... 168
C.2. Comparison of CRISPRDetect with existing CRISPR prediction tools ......... 172
C.3. Examples of the effects of various modules. ...................................................... 174

DVD table of contents ............................................................................................... 177
D.1. Electronic copy of the thesis .............................................................................. 178
D.2. The perl programs for Chapter 2 ....................................................................... 178
D.3. The MySql table for Chapter 2 ......................................................................... 178
D.4. The perl programs for Chapter 3 ....................................................................... 178
D.5. Other executables for Chapter 3 ....................................................................... 178
D.6. Supplementary files for Chapter 3 ................................................................... 178
D.7. The perl programs for Chapter 4 ....................................................................... 179
D.8. Supplementary files for Chapter 4 ................................................................... 179
D.9. The perl programs for Chapter 5 ....................................................................... 179
D.10. Supplementary files for Chapter 5 ................................................................... 179

Glossary ....................................................................................................................... 181
## List of tables

Table 3.1 Summary of features of CRISPR/Cas systems ............................................. 64

Table 3.2 Predicted CRISPR arrays in *Pectobacterium* species ................................. 72

Table 4.1 Presence of ATTGAAA- like motifs at the 3’ end of the array ...................... 89

Table 5.1 The length distribution of the representative repeats in CRISPRDetect output. ........................................................................................................................................... 121

Table A.1 MySQL schema of the table UTR_Sequences .............................................. 158

Table A.2 MySQL schema of the table Predicted_Targets_Info. .................................. 158

Table B.1 Combinations of analysis (leaving out each in turn) for 100 random arrays. ........................................................................................................................................... 164

Table C.1 DR length distribution from (Kunin et al., 2007). ........................................ 170

Table C.2 Spacer length distribution. ............................................................................. 170

Table C.3 Comparison of four widely used CRISPR prediction tools with CRISPRDetect ........................................................................................................................................... 173
List of figures

Figure 1.1 Structure of a typical eukaryotic mRNA.......................................................... 20
Figure 1.2 Schematic representation of a RNA-guided engineered nuclease (RGEN)....... 22
Figure 1.3 Translational control in 3’ UTRs...................................................................... 25
Figure 1.4 Secondary structure and sequence conservation of 2 IRE families .............. 27
Figure 1.5 Overview of CRISPR/Cas adaptive immunity.................................................. 34
Figure 1.6 Comparison of type I, II, and III CRISPR systems........................................ 38
Figure 2.1 Skeleton view of the main modules and steps involved in a SFM analysis... 45
Figure 2.2 The input section of Scan for Motifs ................................................................. 49
Figure 2.3 Scan for motifs interactive graphical output for the TNF gene ................. 53
Figure 3.1 Flowchart of the steps in CRISPRTarget............................................................ 60
Figure 3.2 CRISPRTarget input......................................................................................... 63
Figure 3.4 Graphical output of CRISPRTarget.................................................................. 68
Figure 4.1 Overview of the methods used in CRISPRDirection........................................ 86
Figure 5.2 Extending CRISPRs. A) Initial state of the CRISPR........................................ 115
Figure 5.3 Handling insertion(s) and deletion(s) in CRISPRDetect................................. 117
Figure 5.4 Comparison of the number of CRISPR predicted by four programs......... 120
Figure 5.5 CRISPRDetect predicted arrays from Escherichia coli............................... 122
Figure B.2 CRISPRMap output for 120 unique repeat sequences.............................. 162
Figure B.3 Prediction leaving out specific methods......................................................... 163
Figure B.4 Relative AT richness in the two flanks......................................................... 165
Figure C.1 Fixing gaps at the repeat ends................................................................. 174
Chapter 1

Introduction

Most living cells contain every gene but not all genes are expressed in each cell type. This is partly controlled by small molecular switches in gene expression and is critical for health and diseases, particularly genetic diseases (Ling, et al., 2013; Qureshi, et al., 2010; Scheuermann and Boyer, 2013). Some of these switches are highly conserved across species (Dominski, et al., 2013; Serganov and Nudler, 2013; Soukup and Soukup, 2004).

Messenger RNAs (mRNAs) are translated into proteins that perform many cellular functions. However, on average, about 40% of a mRNA does not actually code for protein. These non-coding untranslated regions (UTRs) within mRNAs are the sites of most regulatory information. This information can determine how and where the proteins are made (Tucker and Breaker, 2005).

Whereas ~75% of the human genome is transcribed into RNA, only ~3% is transcribed into protein coding mRNA (Djebali, et al., 2012). There are about 30,000 protein-coding genes, and possibly as many non-coding RNAs (Lander, 2011; Wright and Bruford, 2011). The discovery of these increasing numbers of non-coding RNAs (ncRNAs) in the last few years has revolutionised our understanding of genes and genome complexity in vertebrates, especially humans (Ling, et al., 2013; Mattick and Makunin, 2006).

One class of such small single stranded evolutionally conserved eukaryotic non-coding RNA molecules are microRNAs (miRNAs). Mature miRNAs bind to specific mRNA sites (6-8 nucleotides long and are partially complementary, known as miRNA binding sites or target sites) and regulate stability and/or translation of protein coding mRNAs. The key regulatory role for mature miRNAs in the post-transcriptional stage includes translational repression, mRNA cleavage and affects mRNA stability (Chen
and Rajewsky, 2007; Moore, et al., 2013). As of yet, miRNAs are the most studied small ncRNA molecules, found to have distinct role in various human diseases including cancer (Calin, et al., 2002; Ciafre and Galardi, 2013; Ling, et al., 2013).

**Figure 1.1 Structure of a typical eukaryotic mRNA.** A mature mRNA contains a 5’ cap, 5’ UTR, coding region, 3’ UTR and a PolyA tail at the 3’ end (modified from Wikipedia).

In parallel to the miRNAs, the RNA binding proteins (RBPs) are also key regulators of mRNA translation at the post-transcriptional level. They can stimulate and inhibit gene expression by sequence-specific interaction with 3’ UTRs (Kedde, et al., 2010; Kloosterman and Plasterk, 2006; Voorhoeve and Agami, 2007). For long time prior to the advancement in miRNA studies, RBPs were considered to be the main regulators of mRNA stability and translational efficiency (Ho and Marsden, 2014).

The functional cooperation between miRNAs and RBPs and its effect on post-transcriptional mRNA expression and/or function has grown in interest in recent years. A number of riboswitches have been found to be involved in miRNA induced post-transcriptional gene silencing. Some of these switches can simultaneously affect both miRNA target sites and recognition elements in 3’ UTRs that bind to RNA binding proteins (Dethoff, et al., 2012; Kedde, et al., 2010). Recent publications have shown evidence that specific miRNAs and RBPs work together to influence transcript decay (Jacobsen, et al., 2010; Jiang and Coller, 2012; Jiang, et al., 2013). Section 1.1 introduces different computational methods and tools widely used for analysis of regulatory elements in eukaryotic RNAs especially non-coding RNA sequences.

In eukaryotes, the role of non-coding RNAs (ncRNAs) in post-transcriptional gene expression especially RNA interference (RNAi), is now well-studied. Non-coding RNAs are also being used for genome editing (Chitwood and Timmermans, 2010;
In genome editing, specific gene(s) in a target genome may be removed or modified using different approaches. Established methods include inserting a short stretch of synthetic DNA obtained by molecular cloning of the genetic material of interest and by using programmable/engineered nucleases\(^2\) (EEN) such as zinc finger nucleases (ZFNs) (Dreyer and Cathomen, 2012; Urnov, et al., 2010), transcription activator-like effector nucleases (TALENs) (Sommer, et al., 2014; Sun and Zhao, 2013), and RNA-guided endonucleases (RGENs) (Sakuma and Woltjen, 2014) that are derived from the prokaryotic adaptive immune system known as CRISPR-Cas system\(^3\).

All these nucleases cleave chromosomal DNA in a targeted manner and produce site-specific DNA double-strand breaks (DSBs) (Chapman, et al., 2012). Through homologous recombination (HR) or non-homologous end-joining (NHEJ), the repairing of these DSBs leads to gene correction, disruption, addition, and targeted chromosomal rearrangements (Brunet, et al., 2009; Moehle, et al., 2007; Sung, et al., 2013; Urnov, et al., 2005). Programmable nuclease-based genome editing is now becoming widely used for making gene knockout/knock-in animals and plants as well as genome-modified cell lines (Kim and Kim, 2014).

Techniques for making targeted nucleases that will recognize and cleave related sequences in genome was initially attempted through changing the specificity of naturally occurring restriction enzymes such as meganucleases (Cohen-Tannoudji, et al., 1998; He, et al., 2014). The meganucleases, also known as homing endonucleases are a class of highly efficient sequence-specific enzymes with long recognition sites (typically 20-30 nt), that can be engineered for targeted genomic modifications (Paques and Duchateau, 2007; Stoddard, 2011).

The zinc-finger nuclease (ZFN) was the first engineered endonuclease (EEN) developed for targeted genome modification. A ZFN is a chimeric protein composed of a functional domain comprising several zinc-finger (ZF) modules and a FokI nuclease

---

\(^2\) A nuclease is an enzyme capable of cleaving the phosphodiester bonds between the nucleotide subunits of nucleic acids.

\(^3\) CRISPR (clustered regularly interspaced short palindromic repeats)/Cas (CRISPR associated): is a bacterial and archaeal defence mechanism against invading foreign nucleic acids.
domain (Carlson, et al., 2012). A later addition to the EEN family are the TALENs, which are structurally very similar to the ZFNs but generally more accessible to the researchers. TALENs are chimeric proteins composed of a FokI nuclease domain and a DNA-binding module called transcription activator-like effectors (TALEs) (Sakuma and Woltjen, 2014).

A new system that has been developed in the last few years is RGENs. The working principle of the RGENs mediated genome editing are based on the type II CRISPR/Cas system observed in prokaryotes. During viral infection, bacteria and archaea often acquire small DNA fragments (known as spacer sequence) from the invading organism, embed the foreign DNA into specific region (i.e. CRISPR array), produces dualRNA-Cas protein complex which identify and cleave the foreign DNA (Kim and Kim, 2014). A detailed description of the natural mechanism can be seen in Section 1.2.2.

RGENs are ribonucleoproteins that consist of a single chain guide RNA...
(sgRNA) (or a crRNA$^4$-tracrRNA$^5$ complex) and a Cas9 protein (Cho, et al., 2013; Kim and Kim, 2014). These enzymes cleave chromosomal DNA, whose sequence is complementary to the guide RNA. This target is also known as protospacer. As Cas9 remains constant, only the guide RNA component needs to be modified for constructing new sets of genome-editing nucleases. Because of the small size of the Cas9 gene (4.1 kbp compared to a pair of ~6 kbp TALEN genes) and of its simplicity to design and prepare, RGENs provide a crucial advantage over ZFNs and TALENs (Kim and Kim, 2014).

Efficient delivery of the engineered nucleases to their target sites is essential. Multiple methods using transfection of plasmid DNA, purified proteins, in vitro transcribed mRNA and viral vectors (e.g. IDLVs$^6$, AAVs$^7$) are reported to accomplish this in both in vivo and in vitro (Hwang, et al., 2013; Kim and Kim, 2014; Meng, et al., 2008; Niu, et al., 2014).

Successful application of genome editing depends on specificity, this varies with different systems. As, the engineered nucleases not only cleave DNA at the desired target site (on-target), it also cleaves DNA similar to the target sites (off-target). This ‘off-target’ cleavage can significantly reduce the efficiency of on-target mutagenesis and increase cytotoxicity (Urnov, et al., 2010). Unlike ZFNs and TALENs whose DNA-binding proteins can be very specific to the on-target sites, RGENs use complementary base pairing to recognize their target sites, which can differ by several nucleotides from the on-target sites (Cho, et al., 2014; Fu, et al., 2013; Hsu, et al., 2013). The protospacer adjacent motifs (PAM) are essential for target recognition by Cas9, but as the PAM is short (5’-NGG-3’), it is required to identify unique target sites in the genome for increased specificity. The off-target effect can be significantly reduced by optimising the nuclease expression levels (Hsu, et al., 2013), modifying the guide RNA (Cho, et al., 2014), and nickases (Cho, et al., 2014). Recent introduction of several off-target finding tools has provided enhanced support to this cause (Bae, et al., 2014; O’Brien and Bailey, 2014). Chapter 3 of this thesis introduces a novel method for predicting true targets of

---

$^4$ CRISPR RNA (crRNA): a small RNA transcribed from the CRISPR loci that determines the target-sequence specificity of Cas9 RNA-guided endonucleases.

$^5$ Trans-activating crRNA (tracrRNA): a small trans-encoded RNA, first discovered in the human pathogen Streptococcus pyogenes.

$^6$ IDLV: Integrase-deficient lentivirus vector.

$^7$ AAV: adenoviruses and adeno-associated virus.
Prokaryotes such as bacteria use a number of defence mechanisms including masking receptors on the cell surface, abortive infection and CRISPR/Cas systems to prevent bacteriophage infections (Labrie, et al., 2010; Petty, et al., 2007). Analogous to eukaryotic RNAi system, the CRISPR/Cas system is frequently observed in archaea (~90%) and bacteria (~50%). Since the first identification in 1987 by Ishino et. al. in *Escherichia coli* (Ishino, et al., 1987), CRISPR/Cas system has generated widespread interest in recent years (Bikard and Marraffini, 2013; Marraffini and Sontheimer, 2010; Wiedenheft, et al., 2012). As, the better understanding of the prokaryotic CRISPR/Cas system could lead to a more efficient RGEN based system, this PhD study involves analysis of several key aspects of the CRISPR/Cas system. A comprehensive study of the CRISPR/Cas system is shown in the section 1.2 of this chapter. A novel technique for accurate identification of the transcribed strand (Biswas, et al., 2014) is described in Chapter 4, followed by an improved CRISPR array prediction tool and an interactive database in Chapter 5.

1.1 Analysis of regulatory elements in eukaryotic RNA

Gene expression at a post-transcriptional level is generally controlled by small RNA molecules or RNA binding proteins (RBPs) interaction with mRNAs. In eukaryotes it is mainly influenced by the regulatory elements present in the 5’ and 3’ UTR regions. These play a key role in various regulatory functions including mRNA transport, intracellular localization, stability and translational efficiency (Deng, et al., 2013; Hamilton and Davis, 2011; Stevens, et al., 2011).
Figure 1.3 Translational control in 3’ UTRs. Binding of mature miRNA packaged in a functional ribonucleoprotein complex (also known as RNA Induced Silencing Complex or RISC) to specific miRNA binding sites in the presence of other regulatory elements (such as RNA binding proteins) may induce translational repression.

Elements, such as AU-rich element (ARE) and C/CU (pyrimidine) rich elements are identified to be involved in RNA stability and frequently found in mRNA 3’ UTRs (Gillis and Malter, 1991; Ho and Marsden, 2014; Shaw and Kamen, 2012). As typical elements are only 10-50 bases long, finding these small elements in the large human genome (~3 billion bases) is difficult and requires sophisticated computer and experimental tools (Abreu-Goodger and Merino, 2005; Hammann and Westhof, 2007). Based on the nature of the analysis and its function, these tools are categorised and discussed in the following sections.

1.1.1 Handling input sequence data

Since the introduction of next generation sequencing (NGS) techniques, one of the major challenges is handling large amounts of genomic sequences. Not only the size of the input data, which ranges from few kilobytes to terabytes, but also the different types of data made data handling more challenging. The raw sequence data generated by NGS techniques is mostly processed in local computing clusters. However, data sources like the UCSC human genome browser (Kent, et al., 2002) or Biomart (Haider, et al., 2009) contains processed and annotated genomic sequences remotely. These public repositories are widely used by researchers, and it is important to have a simple
interface to directly import data from these sources. Galaxy browser’s (Giardine, et al., 2005) “Get Data” module is a good example how to overcome this problem by allowing direct import from many public repositories, as well as direct sequence upload from ftp servers or local system. Galaxy browser’s numerous data conversion and manipulation tools for most of the commonly used format made the data handling significantly simple. All input data is saved with their appropriate descriptors making future use of the data convenient. Furthermore, having the data type specified, the related programs can easily identify supporting and related data that can be used. Another important feature of the Galaxy browser is its capability of handling gene lists in various commonly used formats including BED (browser extensible data), GFF (general feature format) and MAF (multiple alignment format). The BED format provides a flexible way to define data that can be directly incorporated as a custom annotation track in the genome browser visualisation (Giardine, et al., 2005).

With the vast increase in the size of the input sequences, the redundancy in the data increases significantly. Additionally, to identify false discovery rate (FDR) or e-value estimation, it is often required to generate new sub-sequences from the input sequences. Tools like Travesty (Gagnon and Brown, unpublished) can be used for generating new sub-sequence from the selected sequences based on user specified criteria (e.g. N nucleotide shuffle; where N is a positive integer). The CD-HIT-EST (Li and Godzik, 2006) tool provides a fast way to reduce the redundancy from the inputted sequence.

1.1.2 Identification of regulatory elements in RNA

RNA molecules are involved with variety of functions in the cell including regulation of translation and stability. The functions of RNA molecules are often related to their secondary structure, which can be determined by their sequence compositions. One such secondary structural element is the Iron-Responsive Element (IRE), which is a specific RNA stem-loop motif typically found in either of the 3’ and 5’ UTRs. IREs play important roles in iron metabolism, transport and one of the key factors of translational efficiency for some genes (Kim, et al., 1996; Stevens, et al., 2011; Theil, 1998).
Figure 1.4 Secondary structure and sequence conservation of 2 IRE families (A) and (B). IUPAC codes corresponding to the seed sequences in each family are shown. Conserved bases and pairings are highlighted [modified from (Stevens, et al., 2011)].

Although RNA binding proteins recognize RNA targets in a sequence specific manner, the secondary structure of the binding site also influences the binding event. For a deeper understanding of the function of RNA-binding proteins, it is necessary to identify the binding site and target RNAs.

Many RNA-binding proteins have domains that bind to single-stranded RNA. MEMERIS (Hiller, et al., 2006) searches for sequence motifs in a set of RNA sequences and simultaneously integrates information about its secondary structure. It determines values that characterise the single-strandedness of all potential motif occurrences, and use the values as guide for the motif search towards single stranded regions.

REFINE (Riordan, et al., 2011) explicitly finds target-specific motifs while accommodating position specific probability matrix (PSSM) based models in sequences of mRNAs that are selectively bound by specific RNA-binding proteins (RBPs). The identification of motifs is a two-step process. First, it identifies specific RNAs that contain sequence patterns overrepresented in the input sequence data relative to the whole transcriptome. Then it identifies motifs in these segments using existing tools such as MEME (Bailey, et al., 2006).
The small size of RNA binding protein sites with a high degree of complicated composition with high redundancy makes it difficult to obtain an accurate description of them. Some methods developed for TF binding sites (TFBS) in DNA sequences can also be applied to RNAs. Understanding the complex mechanisms of regulation of gene expression requires characterization of the motifs acting both transcriptionally and post-transcriptionally. FIRE (Finding Informative Regulatory Elements) (Elemento, et al., 2007) is a framework that detects DNA and RNA motifs. FIRE yields mutual information between sequence and gene expression measurements, with minimal heuristics about the background sequence model or the presence of other elements that can affect gene expression as input. It directly evaluates the dependency between the location of a motif in the regulatory region (e.g. 3'UTR) and the expression of the corresponding mRNA.

### 1.1.3 Processing putative regulatory elements

Based on the nature of the analysis, this section is divided into four sub categories namely element Conversion, Comparison, Integration and Matching and weighting.

**Conversion.** It is important to note that not all the tools discussed in the previous section produce output in the same format. Such output can be a matrix, sequence, or sometimes a list of identified elements. Hence, element ‘conversion’ is highly important to handle different formats. STAMP (Mahony and Benos, 2007) supports a wide variety of motif input formats including commonly used position frequency matrices as supported by MEME (Bailey, et al., 2006), TRANSFAC (Matys, et al., 2006), JASPAR (Vlieghe, et al., 2006) or ‘Raw Count’ formats, consensus sequences generated in the IUPAC sequence alphabets, as well as multiple alignments of binding sites. Currently it supports direct import of over 12 commonly used motif-finder tools.

**Comparison.** Highly conserved motifs from a set of sequences may potentially share similar regulatory functions. The element ‘comparison’ functionality includes efficient alignment of motifs using local and global alignment methods. STAMP (Mahony and Benos, 2007) uses ‘Sandelin and Wasserman’ method for obtaining the
statistical significance score (p-value) to avoid length biases when comparing motifs of different lengths. It also supports multiple alignments of motifs by ‘progressive profile alignment’ and ‘iterative refinement’ methods, followed by construction of similarity tree by using Phylip software package (Retief, 2000).

**Integration.** This functionality refers to the construction of a generalised profile from the final multiple alignments of the motifs present in the input data. Then compares the identified motifs to generalised ‘familial binding profiles’ (Sandelin and Wasserman, 2004) or by performing a comparative study with the known motifs available in the commonly available databases like Transterm, UTRSite. Hence, even the unknown motifs, which belong to such profile, may provide meaningful structural and functional information.

**Matching and weighting.** This includes matching a list of identified regulatory elements against a user given sequence or a database. Widely used tools like BLAST (Altschul, et al., 1990) and BLAT (Kent, 2002) can be used for matching identified motifs against the mRNA sequences. BLAST is a very time efficient and straightforward approach for searching pattern similarity but often compromises the sensitivity. It supports input sequences either in FASTA or Genbank (Benson, et al., 2012) formats and the output can be obtained in HTML, XML or plain text format. The output contains detailed information about the potential similar regions of the mRNA sequences to the motifs compared. A similar but more accurate and time efficient tool BLAT (BLAST-like alignment tool) could be suitable for comparing the motifs against the mRNA sequences as well. PatSearch (Grillo, et al., 2003) attempts to detect sequence patterns and structural motifs in the input sequence(s) against any number of PWMs. It supports searching of all cis-acting elements available in UTRSite database (Grillo, et al., 2010) against given mRNA sequences. PatSearch also supports identification of transcription factor’s recognition site using homology information and allows a better understanding of the transcription regulation mechanism by using specific combination of PWMs in the analysis.

The non-coding part of the mRNA (5’ and 3’ UTR) sequence contains regulatory elements that control gene expression at the post-transcriptional level (cis-acting elements). The Transterm (Jacobs, et al., 2009; Jacobs, et al., 2002) database contains a
comprehensive list of (~65) manually curated mRNA elements that are involved in translational control. Elements defined in Transterm can be matched in an input sequence or sequences from the Transterm database. It supports a number of filters for target specific sequence data retrieval. All the elements are well documented and cross-referenced to their source. Another web-application, RegRNA (Huang, et al., 2006) supports identification of functional RNA elements, and now contains over 10 types of functional RNA motifs and elements including RNA editing, splicing and polyadenylation sites (Chang, et al., 2013).

The microRNAs (miRNAs) targets in vertebrate messenger RNA (mRNA) sequences are predicted by identifying conserved complementarity to the miRNA seed (nucleotides from position 2 to 8) sequence, which is often anchored by adenosines (A). As the seed region is very small, any typical motif search would generate thousands of matches. Comparative sequence analysis of multiple vertebrate (human, mouse, rat, dog and chicken) 3’ UTRs in alignments revealed that over one third of the human genes are conserved miRNA targets (Lewis, et al., 2005). The requirement of the seeds being conserved in all the species also drastically reduced the false-positives. Additionally, the requirement of a minimum 7 nt long seed (nucleotides from position 2 to 8) matching can be reduced to 6 nt (nucleotides from position 2 to 7) without compromising the specificity. The method was implemented in the TargetScan webserver (Lewis, et al., 2005). The most recent publication from the same group has reported over 45000 conserved miRNA target sites in human 3’ UTRs when analysed with additional vertebrate genomes and an improved background conservation model. Number of improvements related to the background model was reported over previous version. This includes obtaining expected conservation for seeds by comparing the GC content of the matching seed against control seeds, construction of seed-match classes by subtracting the signal and the background of larger seed matches (e.g. 8 nt seed) from smaller seed matches (e.g. 6 nt seed) and construction of UTR specific background conservation using phylogeny (Friedman, et al., 2009).

Several databases (Amaral, et al., 2011; Szymanski, et al., 2003; Szymanski, et al., 2007; Wu, et al., 2006; Zhang, et al., 2010) are available to provide organized information about regulatory noncoding transcripts from different groups of organisms. ncRNAdb (Szymanski, et al., 2007) contains over 30,000 noncoding RNA sequences
from more than 99 organisms. A more specialized database Rfam (Griffiths-Jones, et al., 2003), provides a collection of multiple sequence alignment as well as covariance models of noncoding RNA families, which contains both RNA genes and cis-regulatory mRNA elements. The most recent update of the Rfam (v11.0) database (Burge, et al., 2013), contains over 2208 families with a coverage of over 6 million sequence regions. This makes Rfam the most comprehensive source of ncRNA annotation and functional non-coding RNAs. UTRdb contains a collection of eukaryotic 5’ and 3’ UTR sequences and interlinked with UTRSite, which holds experimentally verified UTR specific elements (Grillo, et al., 2010). Another database, fRNAdb (Kin, et al., 2007) contains a list of putative non-coding RNAs (ncRNAs) pre-computed and annotated from many publicly available databases including H-Inv (Tanino, et al., 2005), NON-CODE (Liu, et al., 2005) and RNAdb (Pang, et al., 2007; Pang, et al., 2005).

1.1.4 Mapping regulatory elements and genes

Noncoding RNAs (ncRNAs) have pivotal roles in gene expression (Rinn, et al., 2007). In some cases, cis-regulatory RNA structures directly regulate translation. Both ncRNAs and regulatory RNA structures are often found to have multiple instances in different genes across a genome. A genome-wide study of these properties can lead to a better understanding of the functional and structural characterization of these regulatory elements. Computational methods like EvoFold (Parker, et al., 2011), successfully demonstrates this for many known families, and predicts many new ones.

Programs designed for eukaryotic ncRNA may be useful in mRNA analysis. Genome-wide computational screening of ncRNAs is a challenging task, as ncRNAs do not share common signals, which can be identified at the sequence levels. However, a large class of ncRNAs has been found to have well conserved functional structures with specific patterns in underlying sequences, and comparative computational approaches RNAz (Gruber, et al., 2010; Hofacker, et al., 1998; Moulton, 2005; Rivas, et al., 2001; Washietl, et al., 2005) has been developed to discriminate functional RNAs from other type of conserved sequences. A large number of noncoding RNAs have been identified in the human genome by mapping the conserved RNA secondary structures against the genome (Washietl, et al., 2005) using tools like RNAz and the UCSC Genome Browser.
1.1.5 Visualisation

Visualisation of both sequence and structures are of great importance for a better understanding of the effectiveness of algorithms, computational tools as well as structure prediction or comparison. The structures may consist of sequences with hundreds of bases and a meaningful visual representation is often a challenging task. Tools like GBrowse (Donlin, 2009), UCSC Genome Browser (Kent, et al., 2002), ncRNA software tools (Asai, et al., 2008) and SynView (Wang, et al., 2006), provide some assistance.

GBrowse (Donlin, 2009) displays genomic sequence features for any known organism, and one of the most actively developed projects. Many software tools like SynView (Wu, et al., 2006) use the GBrowse framework to provide visualization of multi-species comparative genomic data.

The human genome browser at UCSC (Kent, et al., 2002) is another commonly used tool to display any requested portion of the human genome at any scale with several aligned and annotation tracks including assembly contigs, gaps, cross-species homologies and transposon repeats. Combo (Engels, et al., 2006) is another comparative genome browser that provides a dot-plot view of genome alignments synchronized with genome annotations.

TargetScan (Friedman, et al., 2009) provides a comprehensive resource to visualise conserved miRNA targets for ten mammalian species including human with different level of conservation and seed lengths (Garcia, et al., 2011).

The software tools provided by ‘Software.ncRNA.org’ (Asai, et al., 2008), greatly assists visualization of secondary structural elements in alignments. Other online tools such as AURA (Dassi, et al., 2012) can be as well used for showing a number regulatory elements in human 3’ and 5’ UTR sequences.
Structures predicted by different computational tools, are often required to be converted to a more commonly used format prior using the tools. Once such format, called Browser Extensible Data (BED) format is supported by some of these tools. Chapter 2 of this thesis demonstrates a new development (Scan for motifs) with enhanced visualisation capability (Biswa and Brown, 2014).

1.2 Analysis of CRISPR/Cas systems in prokaryotic RNA

Viruses that infect bacteria, bacteriophages (or phages) are abundant in all natural environments and pose a constant threat to its inmates specially archaea and bacteria. In addition to virus attack, there are other biotic threats from toxins and transmissible genetic materials like plasmids, as well as abiotic stresses like non-optimal temperatures or nutrient levels. Also, viruses have a high rate of mutations and recombination. Hence, to successfully defend against these incoming threats, both archaea and bacteria need sophisticated defence mechanisms, which can rapidly adapt to changing threats (Bhaya, et al., 2011; Richter, et al., 2012).

The CRISPR/Cas system is a multistep heritable ‘adaptive immune system’ with a genetic memory of past infections (Terns and Terns, 2011; Wiedenheft, et al., 2012). It has mainly been investigated as a defense mechanism against incoming foreign DNA but is also believed to play important regulatory roles in conjunction with specific Cas proteins in host cells (Bhaya, et al., 2011).

1.2.1 Elements associated with the CRISPR/Cas system

Like other complex systems, the CRISPR/Cas system also contains a range of elements directly or indirectly related to the working of the system. The following section contains a short description of the major elements.
Repeats or direct repeats. Short pieces of repeating sequence from the host, which is mostly identical throughout a CRISPR array. However, mutations may be observed especially at the 3’ end of the CRISPR. The average length distribution of repeats are 23-47 bp. CRISPRs that belong to the same subtype often share identical repeats (Kunin, et al., 2007; Makarova, et al., 2011). Although, the naming of the system suggests that the repeats are palindromic, and in some system it is found to be the case (Mojica, et al., 2000), they are mostly non-palindromic. They may or may not contain a secondary structural element (Kunin, et al., 2007; Makarova, et al., 2011). The foreign DNA sequences (i.e. the spacers) are separated by these repeats (Fig. 1.4).

Spacer. Short pieces of foreign DNA sequence adopted from invading organism (e.g. bacteriophages) or other transferable elements like plasmids (Mojica, et al., 2005; Pourcel, et al., 2005), mostly unique in sequence and grant the hosting CRISPR system sequence specific immunity against its source (Deveau, et al., 2008). Sometimes, spacers are found in other regions of the host genome due to accidental incorporation.
In contrast to the repeats, the length of the spacers may, or may not, be the same length. There are CRISPRs with identical or near identical spacer lengths (e.g. 32) as well as CRISPRs with large difference in spacer lengths (Grissa, et al., 2007).

**Leader sequence.** The upstream of the first repeat at the 5′ end is called the leader sequence and is commonly AT rich. A typical leader can be 200-500 nucleotides long and it is important for new spacer acquisition (Agari, et al., 2010; Jansen, et al., 2002; Pougach, et al., 2010; Richter, et al., 2012).

**Protospacer.** The complementary strand of the spacer sequence in an invading foreign DNA is called protospacer, also known as the targets of the CRISPRs. Any incoming DNA with near identity to the protospacer may be destroyed by the CRISPR interference mechanism (Labrie, et al., 2010).

**PAM.** As the name suggests, the protospacer adjacent motifs (PAM) are located immediately adjacent to the protospacers, 2-5 nucleotides long and have been identified to play functional roles in spacer acquisition and interference (Shah, et al., 2013).

**CRISPR associated (Cas) genes/proteins.** A group of genes mainly located near the CRISPR locus/array that encode proteins associated with CRISPR. There are about 45 Cas proteins identified and out of these 45 Cas proteins Cas1 to Cas4 are found to be specifically associated with CRISPR elements (Haft, et al., 2005; Scholz, et al., 2013). As the Cas1 protein is found in all CRISPR systems, they are recognised as a universal marker of CRISPR/Cas systems. Cas3, Cas9 and Cas10 proteins are specific to CRISPR type I, II and III respectively (Makarova, et al., 2011).

**crRNA.** A CRISPR RNA (crRNA) is a short noncoding RNA composed of repeat and spacer sequences, matured from a pre-crRNA and a key component in the CRISPR/Cas system. It guides the Cas proteins to invading foreign DNA (Chylinski, et al., 2013; Karvelis, et al., 2013; Reeks, et al., 2013).

**CASCADE.** CRISPR-associated complex for antiviral defense (CASCADE) is a ribonucleoprotein complex comprising multiple Cas proteins (Jore, et al., 2011).
Experimental results showed that maturation of pre-crRNA to small functional crRNA depends on a complex of five Cas proteins (CasA, CasB, CasC, CasD and CasE). These mature crRNA guide the CASCADE to specific invading foreign DNA (Brouns, et al., 2008).

Handle. Mature crRNAs are composed of an antiviral spacer sequence, flanked by short repeat fragment on either side. These short repeat fragments are termed as 5’ and 3’ handles based on the side and predicted to serve as conserved binding sites for the CASCADE protein subunits (Brouns, et al., 2008; Richter, et al., 2012).

1.2.2 Mechanism of action of the CRISPR/Cas defence system

The CRISPR system consists of three major stages as shown in the Fig. 1.4 namely i) Adaptation or acquisition of spacers (short single stranded foreign DNA sequence) sequences (Deveau, et al., 2010; Garneau, et al., 2010; Marraffini and Sontheimer, 2010), ii) Expression and generation of crRNA and iii) Interference or immunity (Deveau, et al., 2010; Garrett, et al., 2010; Richter, et al., 2012).

i) Adaptation or acquisition of spacers. Supported by many experiments on both archaea and bacteria including Escherichia coli (Brouns, et al., 2008), Xanthomonas oryzae (Pougach, et al., 2010), Thermus thermophilus (Agari, et al., 2010), and Streptococcus thermophilus (Barrangou, et al., 2007; Deveau, et al., 2010), it is now proven that new spacers are adapted/acquired from invading phages and plasmids at the leader (5’) end between the first repeat (existing) and a duplicated copy of existing first repeat (the new repeat become the first repeat at the leader end). The spacer acquisition process can be subdivided into two groups, a). Naïve acquisition, and b). Priming acquisition. In Naïve acquisition Cas1 and Cas2 are required in duplication of the first repeat and acquisition of the new spacer (Yosef, et al., 2012). Priming acquisition occurs after the initial ‘naïve acquisition’ of the spacer and requires Cas1, Cas2, Cas3, Cascade and a crRNA with sequence specific to the invading foreign DNA for priming and incorporation of the initial spacer (Datsenko, et al., 2012; Swarts, et al., 2012).

ii) Expression and generation of crRNA. After the initial spacer acquisition the
CRISPR locus/array is expressed as a long pre-crRNA. Unidirectional transcription found to occur on the 5’ leader end upstream of the first repeat (Agari, et al., 2010; Pul, et al., 2010). However, bidirectional transcription can be found in archaea (Lillestol, et al., 2006; Lillestol, et al., 2009). The pre-crRNA is then processed by endoribonuclease such as Cas6 (Carte, et al., 2008; Haurwitz, et al., 2010) and cleaved into multiple smaller units (crRNAs) containing a single spacer with a flanking partial repeat sequence. The mature crRNAs typically contains a spacer and 8 nt 5’ handle (Brouns, et al., 2008).

iii) Interference or immunity. Immunity is achieved by forming a ribonucleoprotein complex (crRNP) using the mature crRNAs (used as a guide) and system specific Cas proteins, which binds to specific incoming foreign DNA and neutralize them (Brouns, et al., 2008; Perez-Rodriguez, et al., 2011).

1.2.3 Types of CRISPR/Cas system

CRISPR/Cas systems are divided into three major types termed as type I, type II and type III. Each type consist of multiple subtypes (e.g. I-A, II-A, II-A etc.) based on the associated signature Cas proteins and other specific Cas proteins.

Type I CRISPR systems are comprised of the ‘large multidomain’ signature Cas3 and the conserved Cas1 and Cas2 proteins. Cas3 protein has been found to have unique DNA nuclease and helicase functionalities (Bhaya, et al., 2011; Sinkunas, et al., 2011). Additionally, there are other Cas proteins involved in the formation of the CASCADE-complex and function in the CRISPR interference mechanism (Brouns, et al., 2008). Type I systems are one of the most well studied in E. coli (Jore, et al., 2011) and Pseudomonas aeruginosa (Wiedenheft, et al., 2009) system, and further subtyped to I-A to I-F.
Figure 1.6 Comparison of type I, II, and III CRISPR systems. (A) Expression of pre-crRNAs through transcription; (B) Maturation of pre-crRNAs to crRNAs; (C) Target binding and degradation of foreign DNA/RNA using crRNP and (adapted from (Richter, et al., 2012)).

Type II CRISPRs are associated with the large multifunctional signature Cas9 protein with the ability to produce crRNAs and defuse invading foreign DNAs (Deveau, et al., 2010). The production of mature crRNAs requires short trans-activating CRISPR RNAs (tracrRNAs) together with Cas9 and RNase III (Beloglazova, et al., 2011). These tracrRNAs found to have a near identical (one mismatch) 25 bp long sequence complementary to the repeats of the CRISPR (Deltcheva, et al., 2011). Type II systems have two subtypes II-A and II-B, where II-A are associated with Csn2 proteins and II-B systems includes Cas4 proteins. Type II systems are only found in bacteria.

Type III CRISPRs systems are associated with the signature Cas10 protein that is believed to play key role in mature crRNA production and cleaving foreign DNA
(Anantharaman, et al., 2010; Bhaya, et al., 2011). The expression and interference mechanisms of type III CRISPR systems are quite similar to type I systems, which also requires Cas6 gene for crRNA maturation. The notable difference is, in the type III CRISPR system the endoribonuclease Cas6 binds to the first base of the repeat instead of binding to the repeat stem-loop as found in type I systems, followed by a number of intermediate steps before mature crRNAs are produced (Carte, et al., 2010; Carte, et al., 2008). Type III CRISPR/Cas systems has two subtypes, namely III-A and III-B. The type III-A found to target mRNAs during CRISPR interference in *Pyrococcus furiosus* (Hale, et al., 2008), and type III-B targeting DNA in *Staphylococcus epidermidis* (Marraffini and Sontheimer, 2008).

1.2.4 CRISPR bioinformatics

Considering CRISPR research is still relatively new, a number of bioinformatics tools and databases have been published over the last few years, facilitating identification and annotation of CRISPR/Cas systems. Based on the nature of analysis, the bioinformatic methods can be categorized into three major groups; namely i) Prediction, ii) Annotation, and iii) Target identification.

i) **CRISPR prediction.** CRISPRs were initially treated as genomic repeats or tandem repeats and identified by repeat finding methods like PatScan, REPuter or tandem repeats finder (TRF) (Benson, 1999; Dsouza, et al., 1997; Edgar and Myers, 2005; Godde and Bickerton, 2006; Guan and Zhao, 2005; Jurka, 1998; Marsan and Sagot, 2000; Stoye and Gusfield, 2002). The typical output consists of a list of repeats found in the target genome, and the overall structure of the CRISPR locus/array was not readily available in the output. Additionally, CRISPR repeats may undergo mutation, insertion or deletion, making one or more repeats dropping below 100% identity against the CRISPR consensus/representative repeat. Hence, CRISPR specific prediction tools were highly desirable, and in 2007 three of such tools were published in quick succession.

PILER-CR (Edgar, 2007), an extension of the genomic repeat finding tool PILER (Edgar and Myers, 2005) and a command line executable program, was able to
provide good sensitivity and specificity. The PILER-CR reporting format is easy to read and informative, especially for finding mismatches and insertion/deletions in the repeats. Shortly after the release of PILER-CR, another interactive webserver called CRISPRFinder (Grissa, et al., 2007) was published. CRISPRFinder shown significant improvement over its predecessors PatScan, REPutter and TRF, in identifying shorter CRISPRs (with just 2 or 3 repeats), trailing degenerated repeat(s), handling tandem repeats and joining closely positioned CRISPR arrays.

CRISPRFinder also provided a handful of post-processing tools for additional analysis. Although CRISPRFinder claimed more accuracy in predicting the degenerated repeats at the end of the CRISPRs, it lacks the functionality of representing insertion(s) or deletion(s) in the repeats. The final report only shows the consensus repeat, compromising valuable evolutionary information that lies within the degenerated repeats or repeats with mutation(s).

Another command line executable CRT (CRISPR Recognition tool) was published with demonstrated improvement in CRISPR prediction over existing tools like PatScan and PILER-CR (Bland, et al., 2007). Like CRISPRFinder, CRT also successfully predicted shorter CRISPRs, and degenerated repeats at the end of CRISPRs. However, CRT offers fewer control parameters compared to PILER-CR and CRISPRFinder as well as option to join closely spaced CRISPR arrays. Chapter 5 of this thesis describes an improved algorithm for predicting CRISPR arrays.

ii) CRISPR annotation. The first public database, CRISPRdb (Grissa, et al., 2007), was launched in 2007 by Grissa et al. It contains all repeats and spacers from the entire bacterial and archaeal complete genomes, specifying the position of the repeats and spacers in the genome. CRISPRdb used CRISPRFinder for the initial CRISPR prediction, complemented with many additional tools aiding further analysis including comparison of different CRISPRs, repeat/spacer or Cas gene database lookup and BLAST screening using repeat/spacer/flanking region(s).

As yet, together CRISPRFinder and CRISPRdb have achieved highest number of citations (~315) over any other CRISPR related tools or resources. Another webserver CRISPI (Rousseau, et al., 2009), contains precomputed CRISPRs from all
published archaeal and bacterial genomes, using a minimum 60% repeat identity with the consensus. Neighbouring Cas genes are predicted using Hidden Markov Model (HMM) profiles that are built from the genes present in the genome. Annotation of the different CRISPR specific elements (repeats, spacers etc.) can be obtained by selecting any repeat or spacer from the database.

However, the most widely used publicly available microbial RefSeq genomes hosted at National Center for Biotechnology Information (NCBI) does not annotate CRISPR elements, and CRISPR annotations of prokaryotic genomes are mainly done by DOE-JGI (Department of energy, Joint genome institute) and JCVI (J. Craig Venter Institute) (Mavromatis, et al., 2009; Tanenbaum, et al., 2010). DOE-JGI uses both PILER-CR and CRT for predicting the CRISPRs and the longer CRISPR is selected for annotation. JCVI uses PILER-CR and CRISPRFinder for repeat and spacer annotation (Mavromatis, et al., 2009).

iii) Target identification. A CRISPR may not be exclusive for only one type of invading DNA source like a virus or plasmid, and a typical CRISPR contains spacers from multiple such sources. Targets of a CRISPR, the source of the invading foreign DNA are predicted by analysing the spacer sequences found in the CRISPR. As yet, BLAST screening using spacers against viruses and plasmids has been the most practiced and successful method. Web services like CRISPRFinder (Grissa, et al., 2007), CRISPI (Rousseau, et al., 2009), provide users with the option to select one or all spacers from a specific array and perform a BLAST against the NCBI NT database.

However, a normal BLAST screening may not be so effective, especially when there are spacers from different sources and lengths as short as ~32 nt. Generally, such a BLAST screen will result in thousands of BLAST hits, making it harder to find the true positives. As, the PAMs are important to recognize the target in both the type I and type II system, and mismatches in the handle are essential for type III system, the BLAST hits need to be processed to find potential true positive targets. Often, homologous sequences from the other sequenced strains of the species shows 100% sequence identity over CRISPR regions. So, additional checking is required to avoid false positive targets. A sophisticated algorithm (CRISPRTarget) for predicting true targets of CRISPRs can be seen in Chapter 4 of this thesis (Biswas, et al., 2013).
1.3 Aims of this work

1. Developing bioinformatics methods to identify and visualise range of *cis*-regulatory RNA elements primarily in the human non-coding 3’ UTRs of RNAs. These may then be used to uncover functional cooperation between different types of elements.

2. Developing a new understanding of prokaryotic CRISPR/Cas systems by developing new algorithms and tools for more accurate CRISPR RNA detection, validation, annotation, and target identification.
Chapter 2
Scan for motifs: a tool for the analysis of combinations of regulatory elements in vertebrate 3’ UTRs

This chapter is reproduced from a published paper (Biswas and Brown, 2014) in BMC Bioinformatics journal. Dr C M Brown acted as the primary mentor and contributed towards improving the quality of the paper.

2.1 Introduction

The untranslated regions of mRNA sequences (UTRs) are the sites of most of the experimentally determined regulatory elements in mRNAs (Chang, et al., 2013; Jacobs, et al., 2009). This post-transcriptional regulatory information determines how, when and where mRNAs are translated. A number of tools and methods have been developed to identify cis-regulatory elements (CREs). However, these tools mainly focused on individual types of regulatory elements (REs) in single sequences and are generally CRE specific, ignoring the presence of other types of REs in the neighboring regions (Garcia, et al., 2011). For example, for the microRNA (miRNA) binding sites predicted by a range of tools there is no specific consideration of nearby protein binding sites. It is known however that for some RE to function, they are affected by the presence of other elements or sequences in the neighboring regions (Jacobs, et al., 2009). Most regulatory elements are quite small and many in-silico predictions have high false positive discovery rates (FDRs). Additional positional information of coexisting CREs could improve the accuracy of predictions.
Some complex RNA elements can be both miRNA target sites and recognition elements in 3’ UTRs that bind RNA binding proteins (Dethoff, et al., 2012; Kedde, et al., 2010). Recent publications have shown evidence that specific types of miRNAs and RBPs work together to influence transcript decay (Jacobsen, et al., 2010; Jiang and Coller, 2012; Jiang, et al., 2013). Whether it is binding of miRNA or protein, gene expression regulators (cis-regulatory elements) only function if the sequence environment favors the event. This highlights the fact why some binding sites are functional (true positives) where many others are not (i.e. false positives) in most in-silico predictions. Hence, it is of high importance to consider the sites of every known element and sequence motif in any such predictions. But this often leads to a greater computational complexity, and therefore is avoided by most prediction programs.

For many studies a specific gene of interest is being analysed. Recently developed systems such as RegRNA 2.0 (Chang, et al., 2013), AURA (Dassi, et al., 2012), ARESite (Gruber, et al., 2011), and UTRdb (Grillo, et al., 2010) have provided some support for this type of analysis. However, the analysis of sequence alignments, a representation of overlapping identified elements, FDR estimation, and the ability to include custom sequence motifs in the analysis, are not currently available in a single tool.

A visual representation for every predicted element per region, together with reference tables/databases, suggesting the significance of these individual elements can greatly improve the true positive prediction. Unfortunately, there are no such systems available now. Scan for Motifs (SFM) is an attempt to achieve this primarily for the 3’ UTR regions of human genomes as well as for any sequence/alignments. SFM visually represents a wide range of regulatory elements such as protein binding sites, targets of conserved miRNA families as well as 6-8 base long miRNA seed sequence targets on 3’ UTR sequence from 23 mammalian genomes in one single run. It also provides simple statistics for each element in every region, which in conjunction to the FDR controls allows users to include/exclude any sets of elements in the experiment.
2.2 Materials and methods

The overall process can be divided in three major categories, namely accepting user input, processing the input, and visualization of the findings. The flowchart (Fig.2.1) shows the different components of Scan for Motifs web-server.

**Figure 2.1 Skeleton view of the main modules and steps involved in a SFM analysis.** The user input sections are in dashed boxes. User selected analyses are executed on demand. TargetScan predictions are also re-mapped to the genomic alignments using PERL scripts (labelled MotifMapper).
2.2.1 Obtaining, cleaning and formatting the data

The complete 3’ UTR human sequences\(^8\) (hg19 database) for RefSeq genes was downloaded from UCSC in multi-FASTA format where each exon was separated. A MySQL database table\(^9\) (Appendix A.1) was created to hold all the sequences along with their RefSeq Gene_ID, gene symbol and species identifier as provided by UCSC.

The human 3’ UTR sequences were scanned using the PatSearch\(^10\) (Pesole, et al., 2000) tool to identify all known regulatory elements available from the Transterm database (Jacobs, et al., 2009). The output has been processed to remove the shorter elements totally encapsulated within 100% identical but larger elements (e.g. AU rich elements, Polyadenylation elements). A false discovery rate control was applied during the search for these regulatory elements, and a tab delimited text file was created to store all the positions where the elements were identified.

The RNA-Binding Protein DataBase (RBPDB) (Cook, et al., 2011), is a collection of experimentally verified RNA binding sites, manually curated from literature. RPBDB contains binding data on 272 RBPs, including 71 that have motifs in position frequency matrix (PFM) format, and 36 sets of sequences of in vivo-bound transcripts from immunoprecipitation experiments. There are several ways to search/use RBPDB; some of them are by providing the name of RBPs or RBDs (RNA Binding Domains) or by specifying the species. However, as the goal is to identify RBPs in unknown user sequence(s), I have included the 71 motifs in PFM\(^11\) format.

Another widely used resource in microRNAs target binding site is miRBase (Griffiths-Jones, et al., 2006), contains published miRNA sequences, annotation data and gene targets. As the aim is to identify and visualise the targets of miRNA binding sites, the mature miRNA sequences\(^12\) were downloaded from miRBase website (ftp://mirbase.org/pub/mirbase/CURRENT/mature.fa.gz), processed (reverse

---

\(^8\) DVD: Chapter2/Materials
\(^9\) DVD: Chapter2/Tables/gene_info.sql
\(^10\) DVD: Chapter2/Scripts/PAT_SEARCH
\(^11\) DVD: Chapter2/Scripts/PFMDir
\(^12\) DVD: Chapter2/Scripts/HSADir/mature.fa
complemented and extracted 8 leading bases) to get a list of 2042 seed sequences (each 8 nucleotides long), and stored in a reference text file\textsuperscript{13}.

The 3’ UTR sequences provided by TargetScan (V.6.2) along with the microRNA-binding site related files (miR Family, Predicted Conserved Targets Info, Conserved Family Info) were downloaded. The ‘UTR_Sequences’\textsuperscript{14} file holds multiple sequence alignments (MSA) of 23 vertebrate genomes with human. The human specific sequences has been extracted and the positional information for the miR-binding sites provided in “Predicted Conserved Targets Info” file\textsuperscript{15} has been checked and corrected (updated where needed) against the latest release of hg19 database (provided by UCSC). A MySQL database table named ‘Predicted_Targets_Info’ (Appendix A.2) has been created to hold the positional information for each of these microRNA binding sites.

A custom perl script\textsuperscript{16} was written and used for checking and updating the positional information as mentioned above. The program uses sequence similarity between the latest release of hg19 (from UCSC) and the UTR_Sequences given in TargetScan website. Most of the cases the sequences were 100% identical. However for some genes the sequences were found to be different in length. The program skipped (rejected) the sequences, where sequences given by TargetScan are longer than UCSC’s sequence and multiple mismatches were found between the sequences, preventing a 100% region similarity. A total of 27 of such cases were identified and the microRNA binding sites specified for these genes were omitted from the list.

2.2.2 Handling user input

The user input is of mainly two types, i) query sequence(s) and ii) query element(s). Fig. 2.2 shows the different input options available in SFM web-server\textsuperscript{17}.

\begin{itemize}
\item \textsuperscript{13} DVD: Chapter2/Scripts/HSADir/has_seeds_tabbed.txt
\item \textsuperscript{14} DVD: Chapter2/Scripts/targetscan_files_and_scripts/UTR_Sequences.txt
\item \textsuperscript{15} DVD: Chapter2/Scripts/targetscan_files_and_scripts/Predicted_Targets_Info.txt
\item \textsuperscript{16} DVD: Chapter2/Scripts/targetscan_files_and_scripts/process_UTR_Sequences.pl
\item \textsuperscript{17} DVD: Chapter2/Scripts/sfm_main.pl
\end{itemize}
i) **Query sequence.** Option 1 in Fig. 2.2 shows the different types of sequence that is accepted in SFM. It supports a standard human gene symbol (i.e. LIN28A) given as source of the query sequence. In such cases relative sequence alignments of 23 vertebrates (including human) will be retrieved from previously processed (see 2.2.1) sequences using the inputted gene symbol and as assigned to be the query sequence. Alternately, the user can input FASTA/multiFASTA/clustalw alignments as well as tabular multiple sequence alignment (MSA) formatted sequences as the query sequence. SFM supports assigning reference sequence when the query sequence has more than one sequence. If a human gene symbol was used to get the input sequence, the reference sequence is assigned to human. In all other cases, the top most sequence is considered to be the reference sequence.

ii) **Query elements.** Option 2.A-E in Fig. 2.2 shows the range of query elements FDR controls available in SFM. All the 77 Transterm elements (option 2.A in Fig. 2.2) are associated with individual frequency of occurrence per thousand bases (referred to as the match score) and can be automatically selected/deselected by changing the cutoff score (shown in the red box in option 2.A in Fig. 2.2). Additionally, users can give their own pattern or sequence motif (e.g. AUUUUUUA), which will be searched along with the other selected elements against the query sequence(s) using PatSearch. Similarly, option 2.B-D (Fig. 2.2) shows the elements from RBPDB, TargetScan and miRBase respectively along with the options to control the FDR. The TargetScan elements are available only when a published human gene symbol is used. Option 2.E (Fig. 2.2) provides users the control to include/exclude elements which are not identified in the reference sequence and applicable where more than one sequence is present in the query sequence.
Figure 2.2 The input section of Scan for Motifs showing the range of supported regulatory elements and FDR controls. For a pre-aligned human 3’ UTR (e.g. TNF-NM_000594) it defaults to searching for over 60 TransTerm regulatory elements with expectations of E-value ≤ 0.175 by chance in typical human 3’ UTR (~1000 nt) (A in Fig) and TargetScan miRNA binding site predictions for ~150 conserved miRNA families (C). In this case the sites for RNA binding proteins with E-values ≤ 1.0 per thousand (B) and miRBase 8mer seeds (D) are also selected.

2.2.3 Processing input

Upon receiving the input, SFM searches for the query elements in independent parallel processes\(^{18}\), where the output from one process is not affected by another process. Irrespective of the input sequence types, all sequences are converted into FASTA format. The associated patterns for each selected Transterm elements and user given pattern(s) or sequence motifs are searched with PatSearch. The RNA binding proteins (RBPs) are searched with MotifLocator (Claeys, et al., 2012) using the PWMS

\(^{18}\) Internal functions executed and managed by sfm_main.pl
downloaded from RBPDB. The TargetScan miRNA binding sites and their position of occurrences are retrieved from the MySQL database table (see section 2.2.1) by using the inputted human gene symbol and mapped on the query sequences. Based on the user given seed length (6, 7 or 8 nucleotides), a list of seed sequences are created taking the initial N bases (N refers to user given seed length) from the 2042 seed sequences (see section 2.2.1). As one seed sequence can be associated to multiple miRNAs in a family, a non-redundant list of seed sequences are created by recording the associated miRNAs. These list of short seed sequences are searched in the query sequence(s) using perl RegEx (regular expressions) and position of occurrences are recorded. Once all the processes are finished, the results from these processes are combined and sent to the output (visualisation) module. 

2.3 Results

The SFM web-server analyses sequences that may be aligned vertebrate UTRs, or user inputted sequences or alignments (Fig. 2.1). Five types of elements are searched for in these sequences.

(i) Regulatory elements from the TransTerm database, which includes relevant UTRSite and ARED elements. This provides a curated collection of CREs that function as translational control elements in mRNAs. The computational models (elements) are selected by the user, and/or filtered on empirically determined background frequencies in a shuffled control set. Matches are identified using PatSearch.

(ii) RBP binding sites represented as position frequency matrices (PFM) from the RBPDB. Matches are identified using MotifLocator with a user specified E-value filter.

(iii) MicroRNA target sites predicted by TargetScan 6.2. TargetScan was chosen as it is widely used, and predicts sites on vertebrate alignments.

19 DVD: Chapter2/Scripts/SFM_MODULES/visualisation.pm
Human miRNAs 6 to 8 base seed sequences using MotifMapper. This simple prediction is intended to allow visualisation of most of the potential miRNA binding sites, including likely false positives, if the user desires to.

User defined patterns in PatSearch format. PatSearch allows searches for simple strings, optionally with mismatches insertions and deletions (e.g. GNGNCC), but also more complex elements (e.g. GCG 3…7 GCG, two GCG separated by 3–7 bases) and RNA secondary structures (e.g. p1 = 10…10 4…7 ~ p1, a ten base stem with a loop of 4–7 bases). A full description of the syntax is presented in the help on the SFM server.

On completion of the individual processes, the results are compiled and presented as an interactive visualisation in layers (Fig. 2.3). As an example, we use the well-studied tumor necrosis factor alpha (TNF) 3’ UTR. TNF is a multifunctional cytokine, it regulates the expression of other genes in inflammation and other processes and its expression is regulated at main steps (Giambelluca, et al., 2013). The TNF 3’ UTR has been shown to be targeted by both proteins and miRNA (Shi, et al., 2012; Vasudevan, et al., 2007) and is a classic example of an ARE containing mRNA. MicroRNAs that are confirmed to target this UTR in mammals are miR-16 (Jing, et al., 2005), miR-19a (Liu, et al., 2011), miR-483 (Ma, et al., 2011), miR-125b (Tili, et al., 2007), miR-130 (Bak and Mikkelsen, 2010), miR-181a (Li, et al., 2013), miR-301(Bak and Mikkelsen, 2010). Unusually, a miR-369-3p containing RNA-protein complex binds to targets within the ARE and activates or represses translation in the cell cycle (Vasudevan, et al., 2007). This ARE may also be bound by the RNABP tristetraprolin (TTP) to repress translation (Qi, et al., 2012).

In the SFM analysis using the settings in Fig 2.2, highlights several types of elements from the TransTerm database (Fig 2.3, yellow): the AU rich element (ARE) is represented by hits from three overlapping descriptions (Background E-value per thousand bases 0.06, 0.12, 0.12 respectively, Fig. 2.3) (Halees, et al., 2008); TNF Alpha Stability and Efficiency Element (E-value 0.000008) (Hel, et al., 1998); and two

---

20 The results from individual processes are managed by sfm_main.pl
descriptions of a Polyadenylation Element at the 3’ end (E-value 0.03, 0.02). These are all present in a similar position in the alignment across vertebrates, and the 9-12 base core ARE (Halees, et al., 2008) is repeated (Halees, et al., 2008). The two predicted stability elements in the TNF 3’ UTR have been verified experimentally (Hel, et al., 1998; Shi, et al., 2012) and the polyadenylation signal has a clear match to the consensus (AAUAAA). In addition a 15-LOX-DICE element is predicted (E-value 0.01) in the same location in only 5 of 17 species. From the information linked from the small ‘i’ to the TransTerm entry it can be found that the 15-LOX-DICE is known to have a role in regulating mRNA stability of mRNAs in early erythropoiesis (Thiele, et al., 1999). This may be a false positive, or a novel finding requiring further investigation.

Three predicted overlapping miRNA binding sites are shown (Fig 2.3, red). Interesting that the miRNA binding sites flank the ARE. Each site links to the family of miRNAs that could bind this seed (e.g. miR181abcd/462) this data is inherited from the TargetScan families and predictions (Garcia, et al., 2011). Included in these predictions are miR-19a, miR-181a, miR-130/miR-301 they have been shown to target these regions in the TNF UTR.

Not predicted with the conservative default SFM parameters are two sites for miR-369-3p within the ARE (Vasudevan, et al., 2007). These could be shown when 7mer miRBase seeds (miR-369-3p, UAUUAUU) are selected that overlaps with the ARE. These miR-369-3p sites are also conserved in the alignment. The TargetScan analysis with 153 'broadly conserved' and 'conserved' miRNA families did not predict this site, as miR-369 is poorly conserved (Garcia, et al., 2011) so they are not shown in the results from this analysis (Fig 2.3 red). However, TargetScan does not predict this known site at all (TargetScan webserver) possibly due to the weak AU base pairing within this site.

Such short matches (6mer, 7mer) should be interpreted with caution, as there are over 4000 possible 7mer seeds from the 2043 mature human miRNA seeds in miRBase. This resulted in over 200 hits in the 17,000 nt TNF UTR alignment. However, most of
these matches are not conserved (not present in a similar locations in the alignment) and can therefore be identified as likely false positives by visual inspection of the SFM output.

![Figure 2.3 Scan for motifs interactive graphical output for the Tumor Necrosis Factor (TNF) gene in human 3’ UTR.](image)

Figure 2.3 Scan for motifs interactive graphical output for the Tumor Necrosis Factor (TNF) gene in human 3’ UTR. Known protein and miRNA sites are detected, and additional predictions are made. The experimentally confirmed and conserved ARE mRNA stability elements are shown in the centre (~710-740, yellow). These are flanked by TargetScan miRNA target predictions (red), miR-19a and miR-181a are known to target these sites. The miR-130 TargetScan prediction almost completely overlaps the miR19 site (left, two intensities of red). Some of the additional predictions include patterns of lower specificity (green and blue) are not conserved and may be false positives (e.g. the KHSRP protein binding site to the left (green ~670), or the miR-150-5p (blue ~760). However miR-125 (blue 8mer seed match ~690) does target this UTR. The results can be downloaded for further study (upper right). See the Results and Discussion section for further analysis. Additional information can be obtained by clicking on the highlighted element.

### 2.4 Discussion

SFM is a free web-application, allowing researchers to use a single tool to identify and
investigate a range of CREs on alignments. Notably, protein binding sites (Transterm, UTRSite, ARED) and miRNA binding sites (TargetScan, seed match) are particularly important in 3’ UTRs. These elements come from well-documented databases and are cross-referenced. The server is a pure LAMP (Linux, Apache, MySQL and Perl) implementation providing speed and stability, using HTML, JavaScript and AJAX to provide seamless user interaction throughout the analysis. SFM visually represents different types of CREs in one display (Fig. 2.3). It also provides the user the choice to include/exclude any sets of elements in the analysis, as well as only showing elements also found in the reference sequence (e.g. human when a gene symbol is used as input). Simple statistics for each of these identified elements can be obtained by mouse clicking on the highlighted elements (Fig. 2.3). Along with the graphical display, SFM also provides a complete text report listing the entire user input (selections and input sequence) as well as results of each individual identification process showing the elements and positions in tab-delimited format. Such a report can be used to perform future analysis revealing association between different types of elements. SFM has been tested on commonly used web-browsers including Chrome, Firefox, Safari and Internet Explorer. SFM will be particularly useful to researchers to easily identify the genomic regions of highly conserved elements and uncover relationships among different classes of post-transcriptional regulatory elements.
Chapter 3

CRISPRTarget: Bioinformatic prediction and analysis of crRNA targets

This chapter is reproduced from published work (Biswa, et al., 2013). I developed the entire CRISPRTarget web-application under Dr C M Brown’s supervision. Some logic was adapted from an earlier unpublished work by Josh Gagnon (Bioinformatics Programmer) and Dr C M Brown. Dr C M Brown also contributed towards the paper, specially the introduction and discussion. Dr P Fineran did extensive and critical testing of the software followed by contribution towards the Pectobacterium specific analysis. Dr Stan J.J. Brouns tested versions of the software and made critical comments on the results.

3.1 Abstract

The Bacterial and Archaeal CRISPR/Cas adaptive immune system targets specific protospacer nucleotide sequences in invading organisms. This requires base pairing between processed CRISPR RNA and the target protospacer. For type I and II CRISPR/Cas systems protospacer adjacent motifs (PAM) are essential for target recognition and for type III, mismatches in the flanking sequences are important in the antiviral response. In this study we examine the properties of each class of CRISPR. We use this information to provide a tool (CRISPRTarget) that predicts the most likely targets of CRISPR RNAs (http://bioanalysis.otago.ac.nz/CRISPRTarget). This can be used to discover targets in newly sequenced genomic or metagenomic data. To test its utility, we discover features and targets of well characterised Streptococcus thermophilus and Sulfolobus solfataricus type II and III CRISPR/Cas systems. Finally,
in *Pectobacterium* species, we identify new CRISPR targets and propose a model of temperate phage exposure and subsequent inhibition by the type I CRISPR/Cas systems.

### 3.2 Introduction

The CRISPR (clustered regularly interspaced short palindromic repeats)/Cas (CRISPR associated) system has evolved to defend microorganisms against foreign invading nucleic acids, principally DNA from bacteriophages (phages), plasmids and other mobile elements (reviewed in (Bhaya, et al., 2011; Richter, et al., 2012; Terns and Terns, 2011; Westra, et al., 2012; Wiedenheft, et al., 2012)). CRISPR/Cas systems have been identified in 47% and 86% of complete Bacterial and Archaeal genomes (Grissa, et al., 2007). Resistance development occurs when a short sequence is acquired from the phage or plasmid genome and added, as a new spacer, to the CRISPR arrays (reviewed in (Fineran and Charpentier, 2012)), which consist of short repeats separated by spacers. In CRISPR systems, a CRISPR RNA (crRNA) containing a ‘spacer’ (or guide (Carte, et al., 2008)) is generated from a longer precursor (pre-crRNA) (Brouns, et al., 2008; Carte, et al., 2008; Deltcheva, et al., 2011; Haurwitz, et al., 2010; Nam, et al., 2012; Przybilski, et al., 2011) and incorporated into a ribonucleoprotein complex of one or more Cas proteins (Brouns, et al., 2008; Gasiunas, et al., 2012; Hale, et al., 2009; Jinek, et al., 2012; Jore, et al., 2011; Lintner, et al., 2011; Wiedenheft, et al., 2011; Wiedenheft, et al., 2011; Zhang, et al., 2012). These ribonucleoprotein complexes bind to, and trigger, the destruction of complementary DNA or RNA from invading elements (Deveau, et al., 2010; Jinek, et al., 2012; Westra, et al., 2012).

Typically, organisms have several CRISPR arrays containing a range of spacers with different sequences derived from previous exposure to phages and plasmids. The largest predicted bacterial array, from *Haliangium ochraceum* DSM 14365, has 587 spacers, only two of which are identical (Grissa, et al., 2007). Despite experimental proof that CRISPR/Cas systems target phages or plasmids (Barrangou, et al., 2007; Cady, et al., 2012; Deveau, et al., 2010; Marraffini and Sontheimer, 2008), the targets of most spacers have not been identified. For example, of 926 spacers identified for *E. coli*
and *Salmonella*, Touchon and Rocha were only able to predict the likely targets of 8% (Touchon and Rocha, 2010). Similarly, a parallel study discovered the targets of 12% of spacers (Diez-Villasenor, et al., 2010).

There are many contributing reasons for the lack of identified crRNA targets. This is partly due to the relative paucity of studies that investigate the sequences of phages when compared with their abundance and genetic diversity (Hatfull and Hendrix, 2011; Krupovic, et al., 2011). Furthermore, many phage sequences are not easily accessible in databases such as GenBank, but many more exist in viral metagenome or virome studies (Culley, 2011). Large proportions of phage sequences have no similarity to any known phage or other sequences. Therefore, most metagenomic data remains unannotated (Hatfull and Hendrix, 2011; Krupovic, et al., 2011). For example, in a recent study the metagenomes of phages purified from thermal ocean vents were sequenced (Ray, et al., 2012). The method targeted lambdoid viruses and resulted in the sequencing of a new lambdoid virus; however, 45-55% of sequences had no database matches (Ray, et al., 2012). Another study of marine viromes identified only 10% of genes related to known phages (Angly, et al., 2006). The lack of identified crRNA targets in plasmids results from a similar dearth of sequence data relative to their abundance and diversity. Like phages, plasmids are mobile and have mosaic sequence structures and are rapidly evolving (Leplae, et al., 2006). Recent efforts have begun to sequence populations of plasmids using metagenomics, which should start to improve this plasmid data shortage (Li, et al., 2012).

CRISPR/Cas systems are divided into three major types (I-III), and further into subtypes (e.g. types III-A and III-B) (Makarova, et al., 2011). Different types share similarities, yet can have differences, such as in crRNA generation or the nature of the target (RNA or DNA). Recent studies have begun to elucidate the process of recognition of target protospacers in the major types of CRISPR/Cas systems. From early studies it was interpreted that exact pairing along the length of the spacer RNA was required (Barrangou, et al., 2007; Deveau, et al., 2008), but recent results indicate that some mismatches are tolerated, at least for some systems (Cady, et al., 2012; Gudbergsdottir, et al., 2011; Manica, et al., 2011; Semenova, et al., 2011). For type I
CRISPR/Cas systems have similarity, yet differences, to RNAi in mammals, which can also provide protection from viral infection (van Rij and Andino, 2006; Wiedenheft, et al., 2012). RNAi utilises miRNAs of ~20-22 bases that recognise specific mRNA targets (Thomson, et al., 2011). However, the key seed determinant is only 6-8 bases. Therefore, predictive tools to discover functional binding sites have been developed that use the properties of known sites to predict new ones (Thomas, et al., 2010; Thomson, et al., 2011). The critical factor is distinguishing true sites from false positives, and there are a large number of algorithms implemented for miRNA target discovery (Saito and Saetrom, 2012; Tan Gana, et al., 2012; Witkos, et al., 2011). A number of bioinformatic tools are available for the identification of CRISPR arrays and their spacer sequences (Bland, et al., 2007; Edgar, 2007; Grissa, et al., 2007). In contrast, few approaches have been developed to discover the targets of CRISPR (Cady, et al., 2011; Grissa, et al., 2007; Rousseau, et al., 2009). For predicted spacers in CRISPRdb (Grissa, et al., 2007) (Nov 2012) the mean length is 36 bases (range:16-100), which is consistent with typical lengths for experimentally confirmed spacers of 24-37 bases. The input used when searching for targets using CRISPRFinder is a small number of these spacer sequences without adjacent repeats (Grissa, et al., 2007). These spacers are used for a BLAST search of the nucleotide sequence database from GenBank using the default parameters (Johnson, et al., 2008; Morgulis, et al., 2008). The discovery of new spacer targets would be facilitated by tools that allow flexibility in these areas and enable searches of recent metagenomic datasets. Furthermore, the ability to score or visualise PAMs, base pairing in flanking sequences, and define seed
regions that are not available in existing tools. These properties assist in biological interpretation of putative targets. In this study we have incorporated known features of the CRISPR/Cas system into a target discovery tool. We have also allowed flexibility to enable the incorporation of new features, to generate testable hypotheses as to the targets of CRISPR systems.

3.3 Materials and Methods

**Target databases.** Selected databases\(^{21}\) are provided in CRISPRTarget. *Genbank databases:* BLAST Nucleotide databases a) The nr/nt collection \~43 billion bases (15/10/2012, Genbank 192). This database contains “All GenBank + EMBL + DDBJ + PDB sequences (but no EST, STS, GSS, or phase 0, 1 or 2 HTGS sequences).” b) env(nt), 8.5 billion bases (15/10/2012). This contains “Sequences from environmental samples, such as uncultured bacterial samples isolated from soil or marine samples. The largest single source is Sargasso Sea project. This does not overlap with nucleotide nr”. This is part of the whole genome shotgun (wgs) but these sequences have no taxonomic classification other than metagenome. c) Phage division (phg). This is one of the smallest GenBank divisions containing 6,800 sequences of 88 million bases. *RefSeq databases:* Several relevant divisions of the NCBI Reference Sequence databases are available, which contain better annotated (by NCBI) versions of GenBank sequences. a) RefSeq-Plasmid. 3,707 sequences, 282 million bases. b) RefSeq-Viral. 4,279 sequences, 95 million bases. c) RefSeq-Microbial. 5,234 complete microbial genomes, 7 billion bases. We also included parts of the CAMERA databases. 913,983 sequences, 1 billion bases. ACLAME. 125,190 sequences, 96 million bases. d) User defined. Users can upload sequences of up to 50 Mb.

**CRISPR array sequences.** CRISPR arrays were used from published studies or CRISPRdb. They were also predicted\(^{22}\) with CRISPRFinder, PILER-CR or CRT using the default parameters. The current tools for prediction have some limitations, notably

---

\(^{21}\) As the databases are big (~50GB) they are not part of the CD. However, the databases can be downloaded from CRISPRTarget website.

\(^{22}\) DVD: Chapter3/Scripts/executables/
the inability to predict the transcribed strand, the imprecise definition of the DR/Spacer junctions, or splitting into several sub arrays.

**Figure 3.1 Flowchart of the steps in CRISPRTarget.** Input is predictions of the CRISPR arrays, selected databases and initial parameters. This input is processed and the spacers screened using BLASTn for matches against the databases. The flanks of these matches are extended and PAMs and handles analysed in an interactive manner. Output is as a text/spreadsheet format, or as a graphical display (HTML).

**Algorithm.** A schematic representation of the CRISPRTarget processes is depicted in the Fig. 3.1.

**Input data-** Spacer sequences are extracted from the input CRISPR arrays using the locations specified and converted to FASTA format. Alternatively, spacer sequences can be uploaded directly, without repeat sequences, however this limits subsequent processing.
**BLAST screen**- Each spacer sequence is used to query the selected databases. Multiple databases can be selected (except where there are identical accession identifiers: nt + phg). The default values used by NCBI BLASTn for short sequences <30 bases (defaults for long sequences are in brackets) are Gap open -5(-5), gap extend -2(-2), match +1(+1), mismatch -3(-3), word size 7(11), Expect (E): 1000 (10). Filter: No (Yes). The initial CRISPRTarget defaults are the same except that a gap is penalised more highly (-10), the mismatch penalty is -1 and the E filter is 1. In addition, there is also no filter or masking for low complexity. The CRISPRTarget BLASTn parameters favor gapless matches but allow a number of mismatches at this screening stage. BLAST calculates the scores over the length of the match, and only shows this match. For example, a spacer of 32 bases that matches to a target in 17 of 20 bases would score 20-3=17 and 20 bases would be output. The expected (E) values of the match will be more likely to pass the filter if smaller databases are used (e.g. the default phg and plasmid). The hits are converted into GFF format.

**Extension of the BLAST match**- The full spacer and handles are extracted from the input sequences. In the case of CRISPRFinder input, only a single repeat is in the input and this is used for all spacer handles. Both CRT and PILER-CR outputs enable small differences in the repeat to be used. If the user wishes to extract more sequence than provided in the array files, e.g. the sequence following the final repeat, this can be extracted from a FASTA file (if provided by the user). Extension of the spacer is not possible if only spacer sequences are in the input. The protospacer target is extended by extracting sequence (using the user-specified length) from the BLAST database.

**CRISPRTarget interactive scoring**- All putative spacer/protospacer targets passing the BLAST screen are displayed in an interactive manner. An initial score is calculated by scoring matches (+1) and mismatches (-1) across the whole length of the spacer without gaps. Specific user defined 'seed' regions can be required to match at either or both ends of the protospacer. A match to predefined, or novel user defined, PAM sequences can increase the score. In order to penalise self-matches that would match 100% in both spacers and flanking handles (e.g. to the original genomic array
sequence), a score can be used that penalises matches (e.g. -1) in the flanking handles. Mismatch penalties can also be used to identify targeting that is facilitated by mismatches in the handles (e.g. type III-A) (Marraffini and Sontheimer, 2010). Finally, a cutoff score can be applied to display only those matches with the best scores.

3.4 Results

**CRISPRTarget: development of a tool for discovery of crRNA targets**

The lack of tools for prediction of the protospacer targets of crRNAs, led us to develop a web application called CRISPRTarget23 (Fig. 3.2). We have summarised the current state of knowledge about the three major CRISPR/Cas types, their PAMs, handles and seed regions (Table 3.1) and used this information when developing CRISPRTarget. Users can provide their input as either spacers in FASTA format, or as CRISPRFinder (Grissa, et al., 2007), PILER-CR (Edgar, 2007) or CRISPR Recognition Tool (CRT) (Bland, et al., 2007) output files (following CRISPR prediction via one of these methods). Putative protospacer targets can be identified, following a BLASTn search of the spacer input against a number of databases or user-uploaded sequences. These databases include ACLAME genes, Genbank-nt, Genbank-Environmental, Genbank-Phage, RefSeq-Microbial, RefSeq-Plasmid, RefSeq-Viral and parts of CAMERA. Although default setting allows the sensitive detection of potential targets, users have the ability to modify the search parameters, such as E-value, word size and penalties for gaps and match/mismatch. This flexibility enables the stringency of targets to be adjusted, depending on the user requirements. Either on the initial input screen, or following the BLAST search, targets can be displayed and scored for flanking sequences, PAMs and filtered by exact matching seed regions. These are important parameters when considering biological details about the predicted target, such as what type of system/CRISPR-type is involved. This information in Table 3.1 can assist users in choosing the appropriate parameters for their particular target search. The output provided is either visual in HTML format, but can also be saved as text and opened in a spreadsheet. The target sequence is typically displayed as an R-loop, depicting a

23 DVD: Chapter3/CRISPRTarget/crispr_target.pl
specified part of the crRNA, as well as both the target and non-target strand of the double stranded target DNA. The target sequence R-loop can be fully reverse complemented, when users suspect that the direction of transcription of the CRISPR array starts from the downstream end instead.

Figure 3.2 CRISPRTarget input. Several formats are accepted. The BLASTn parameters for the initial screen are defined at this step. They default to values that favor a gapless match, but some mismatches. The output may be refined and reordered (Fig. 3.4A) after it is obtained.
Table 3.1 Summary of features of CRISPR/Cas systems including PAMs, repeats, seed regions and handles*

<table>
<thead>
<tr>
<th>Type</th>
<th>Target</th>
<th>Representative species</th>
<th>PAM (5'-3')§</th>
<th>Typical repeat</th>
<th>CRISPR family (Kunin, et al., 2007)</th>
<th>Seed region</th>
<th>5'/3' handles (nt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type I (PAMs 3’ of protospacer)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-A</td>
<td>DNA</td>
<td><em>Sulfolobus solfataricus</em> P2</td>
<td>Protospacer-NGG (Gudbergsdottir, et al., 2011; Lillestol, et al., 2009; Mojica, et al., 2009)</td>
<td>GATAATCTCTTATAGAATTGAA AG¶</td>
<td>CRISPR-7</td>
<td>Unknown</td>
<td>8 / 16-17 (Lintner, et al., 2011)</td>
</tr>
<tr>
<td>I-B</td>
<td>DNA</td>
<td><em>Clostridium thermocellum</em> ATCC 27405</td>
<td>Unknown</td>
<td>GTTTTTATCGTACCTATGAGGA ATTGAAAC¶</td>
<td>CRISPR-6</td>
<td>Unknown</td>
<td>8 / 4, 10-12 (Richter, et al., 2012)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>C. thermocellum</em> ATCC 27405</td>
<td>Unknown</td>
<td>GTTGAAGTGTTACTTCCAGTA AACAAGGATTGAAAC¶</td>
<td>CRISPR-9</td>
<td>8 / 2-6 (Richter, et al., 2012)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Haloferrax volcanii</em> H26</td>
<td>Protospacer-GAA, AGT, TTA, ATA, CTA, GTG (Fischer, et al., 2012)</td>
<td>GTTTCAGACGAACCCTTGTGG GDTTGAAAGC¶</td>
<td>CRISPR-6†</td>
<td>8 / 2-6 (Richter, et al., 2012)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Listeria monocytogenes</em></td>
<td>Protospacer-NGG (Mojica, et al., 2009)</td>
<td>GTTTTTAATCACTTATTATGAAA TCTAAAT</td>
<td>CRISPR-1</td>
<td>8 / 2-6 (Richter, et al., 2012)</td>
<td></td>
</tr>
<tr>
<td>I-C</td>
<td>?</td>
<td><em>Xanthomonas oryzae</em></td>
<td>Protospacer-GAA (Mojica, et al., 2009; Semenova, et al., 2009)</td>
<td>GTCCCGTCTCCTACGCGGCCT GGATTGAAAC¶</td>
<td>CRISPR-3</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Bacillus halodurans</em></td>
<td>Protospacer-GAA (Mojica, et al., 2009)</td>
<td>GTCCGACTCTTCTCATGGGTGCG TGGATTGAAAT</td>
<td>CRISPR-3</td>
<td>11 / 21 (Nam, et al., 2012)</td>
<td></td>
</tr>
</tbody>
</table>

*Continued on next page*
Table 3.1 Continued from previous page

<table>
<thead>
<tr>
<th>Type</th>
<th>Target</th>
<th>Representative species</th>
<th>PAM (5'-3')§</th>
<th>Typical repeat</th>
<th>CRISPR family (Kunin, et al., 2007)</th>
<th>Seed region</th>
<th>5'/3' handles (nt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-D</td>
<td>?</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>I-F</td>
<td>DNA</td>
<td><em>Pseudomonas aeruginosa</em> PA14</td>
<td>Protospacer-GG (Cady, et al., 2012; Mojica, et al., 2009)</td>
<td>GTTCACTGCCGTAGGCGCAGC TAAGAAA§</td>
<td>CRISPR-4</td>
<td>1-8 (Wiedenheft, et al., 2011)</td>
<td>8 / 20 (Haurwitz, et al., 2010; Wiedenheft, et al., 2011)</td>
</tr>
<tr>
<td></td>
<td>DNA</td>
<td><em>Pectobacterium atrosepticum</em> SCR11043</td>
<td>Protospacer-GG (Mojica, et al., 2009)</td>
<td>GTTCACTGCCGTACAGGCCGC TTAGAAA§</td>
<td>CRISPR-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type II (PAMs 5' of protospacer)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Seed adjacent to PAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>II-A</td>
<td>DNA</td>
<td><em>Streptococcus thermophilus</em></td>
<td>WTTCTNN-protospacer (Horvath, et al., 2008)</td>
<td>GTTTTTGTACTCTCAAGATTTA AGTAACTGTACAAC</td>
<td>CRISPR-10</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DNA</td>
<td><em>Streptococcus thermophilus</em></td>
<td>TTYRNNN-protospacer (Bolotin, et al., 2005)</td>
<td>GTTTTTGTACTCTCAAGATTTA AGTAACTGTACAAC</td>
<td>CRISPR-10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Type</th>
<th>Target</th>
<th>Representative species</th>
<th>PAM (5'-3')§</th>
<th>Typical repeat</th>
<th>CRISPR family (Kunin, et al., 2007)</th>
<th>Seed region</th>
<th>5'/3' handles (nt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>II-B</td>
<td>DNA</td>
<td><em>Streptococcus thermophilus</em></td>
<td>CNCCN-protoscaler (Horvath, et al., 2008; Magadan, et al., 2012)</td>
<td>GTTTTAGAGCTGTGTTTTCGAATGTCCAAAAAC</td>
<td>CRISPR-10</td>
<td>13 (Jinek, et al., 2012)</td>
<td>None/19-22 (Deltcheva, et al., 2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Streptococcus pyogenes</em></td>
<td>CCN-protoscaler (Jinek, et al., 2012; Mojica, et al., 2009)</td>
<td>GTTTTAGAGCTATGCTGTTTTGAATGGTCCAAAAC</td>
<td>CRISPR-10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type III (no PAM)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III-A</td>
<td>DNA</td>
<td><em>Staphylococcus epidermidis</em></td>
<td>No PAM (Marraffini and Sontheimer, 2010)</td>
<td>GATCGATACCCACCCCCGAAGAAAAGGGACGGAAC§</td>
<td>CRISPR-8</td>
<td>Unknown</td>
<td>8 / (37/43 entire length) (Hatoum-Aslan, et al., 2011; Marraffini and Sontheimer, 2010)</td>
</tr>
<tr>
<td>III-B</td>
<td>RNA</td>
<td><em>Pyrococcus furiosus</em></td>
<td>No PAM (Hale, et al., 2012)</td>
<td>GTTCCAAATGAATACAAAAATA GAATTGAAAG§</td>
<td>CRISPR-6</td>
<td>Unknown</td>
<td>8 / (39/45 entire length) (Hale, et al., 2012; Hale, et al., 2009)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Sulfolobus solfataricus</em></td>
<td>No PAM (Zhang, et al., 2012)</td>
<td>GATTAATCCAAAAAGGAATT GAAAG§</td>
<td>CRISPR-7</td>
<td>Unknown</td>
<td>8 / uncertain (Zhang, et al., 2012)</td>
</tr>
</tbody>
</table>

*Adapted from Westra et al., 2012
§PAM and protoscaler are denoted as sequence on target strand that base pairs with crRNA
¶Direction of CRISPR transcription confirmed
†Described as CRISPR-9 (Fischer, et al., 2012), but length and sequence suggests degenerate CRISPR-6
Figure 3.3 Example annotated CRISPRTarget outputs of representatives of type I, II and III CRISPR/Cas systems. The protospacer is the DNA target complementary to the crRNA spacer. The crRNA is displayed as RNA 5' to 3' and the base paired protospacer is 3' to 5'. (A) The predicted spacer 6 crRNA from the type I-F CRISPR1 (CRISPR1_6) in *P. aeruginosa* PA14 targets *Pseudomonas* phage JBD67 (Cady, et al., 2012). The output visualizes the 5'-protospacer-GG-3' PAM (Mojica, et al., 2009) and the crRNA with 8 and 20 nt 5' and 3' handles, respectively (Haurwitz, et al., 2010). (B) The CRISPR1_15 from the type II system from *Streptococcus thermophilus* DGCC7710 WTphi858phi2972+S13S14 (Barrangou, et al., 2007) matched to *Streptococcus* phage 5093. The output shows the predicted length of the 3' handle, based on *Streptococcus pyogenes* (Deltcheva, et al., 2011), and the 5'-WTTCCTNN-protospacer-3' PAM (Horvath, et al., 2008). (C) Spacer 1 from the type III-A system from *Staphylococcus epidermidis* RP62a targeting plasmid pGO1 (Marraffini and Sontheimer, 2008). The output was adjusted to display the 8 nt 5' handle with an entire mature crRNA length of 43 nt and no PAMs were scored (Hatoum-Aslan, et al., 2011). Yellow sequences include spacer and protospacer, blue indicates flanking sequences and PAMs are shown in green.

A general model of the match between a spacer and protospacer target as the output from CRISPRTarget is shown in Fig. 3.3 for types I, II and III. The differing features, such as 5' or 3' handles and the presence or absence of PAMs, can be specified, searched, sorted and displayed in CRISPRTarget. Furthermore, the parameters can be manually adjusted to incorporate new functional information (e.g. a new PAM). For clarity, we use the definition of the protospacer as the DNA strand complementary to the crRNA, and PAMs are denoted...
5'-3' on the protospacer DNA (e.g. type I-E PAM is CTT, Table 3.1) (Westra, et al., 2012). In addition, we refer to the flanking sequences as being 5' or 3' of this protospacer and handles as 5' or 3' of the crRNA spacer. CRISPRTarget enables detection of the most likely complements of spacers in target sequences (Fig. 3.4B).

Figure 3.4 Graphical output of CRISPRTarget. The output of a search for the targets of the Streptomyces thermophilus DGCC7710 CRISPR array. The direction of transcription is known; however, both strands are shown in the diagram, as if the direction of transcription was unknown. Two relatively low-scoring matches using these interactive settings are shown (rank 44–45). They have good spacer-protospacer base pairing but lack a WTTCTNN PAM. Match 45 is to a phage to which this strain is sensitive (Φ2972). Yellow indicates spacer/protospacer, blue shows flanking sequences and mismatches between the crRNA and the target DNA protospacer are indicated in red.
Proof of principle: phage protospacers for *Streptococcus thermophilus* type II CRISPR/Cas

As an initial test we used the well-characterised type II CRISPR1 array from *Streptococcus thermophilus* DGCC7710. This strain is economically important in the dairy industry and has active CRISPR/Cas systems (Barrangou, et al., 2007; Deveau, et al., 2008). The sequences of arrays with recently acquired spacers are available WTphi858phi2972+S9S10S11S12 (GenBank accession: EF434477) and, WTphi858phi2972+S13S14 (EF434478), as are many *Streptococcus thermophilus* phage sequences (114 sequences of 6,800 in the phage division of GenBank). These two strains have become resistant to \(\phi858\) and \(\phi2972\), whereas the WT strain is sensitive (EF434469) (Barrangou, et al., 2007). We expect that spacers from the resistant strains will be predicted to target \(\phi858\) and \(\phi2972\), whereas the WT will not, but might target other mobile elements. Spacers were predicted from these CRISPR sequences using CRISPRFinder (Grissa, et al., 2007), CRT (Bland, et al., 2007) and PILER-CR (Edgar, 2007). CRISPRFinder is the most cited CRISPR prediction tool; however, a combination of CRT and PILER-CR are used in the DOE-JGI standard pipeline for bacterial genome annotation (Mavromatis, et al., 2009).

CRT and CRISPRFinder predicted the published array of 32 spacers in the WT and an additional 2 or 4 spacers in the resistant strains, whereas PILER-CR with default parameters split the array into two consisting of 22 and 3 spacers. The CRT predictions were used as input (Fig. 3.2), as these include information about small variations in the repeats (all inputs in this study are provided in Supplement24). These spacers were searched against the phage division of GenBank and plasmid division of RefSeq. With the default settings, there were matches from 24 of 32 spacers to 84 sequences of mobile elements in the initial output, of which 81 were *Streptococcus* spp. phages (Supplement-hmtl25, text26). This has been designated a type II-A system with a requirement for a PAM 5’ of the protospacer (5’-WTTCTNN-protospacer-3’) (Horvath, et al., 2008); 38/84 had the

24 DVD: Chapter3/Supplements/file1.txt
25 DVD: Chapter3/Supplements/file2.html
26 DVD: Chapter3/Supplements/file3.txt
consensus PAM. The additional spacers in strains WTphi858phi2972+S9S10S11S12 and WTphi858phi2972+S13S14 targeted \( \Phi \)2972 and \( \Phi \)858 as expected (Barrangou, et al., 2007). Interestingly, the WT has a spacer (CRISPR1_14, uniquely identified as EF434469_1_14 in the Supplementary text output file S3) with just one mismatch (protospacer +7) to bases 31869-31897 of \( \Phi \)2972. Additionally, the 5' region of the target differs by one base from the PAM consensus (WTcCTNN) (Fig. 3.4 and Supplement27). Experimentally this strain is sensitive to \( \Phi \)2972 (Barrangou, et al., 2007), so the system appears to have a functional requirement for the conserved consensus PAM and/or an exact match near the 5' end of the protospacer, which corresponds to the 13 nt seed region in type II systems (Table 3.1) (Jinek, et al., 2012). In summary, CRISPRTarget can accurately identify protospacers for crRNAs and display these with details of match/mismatch and PAMs.

**Identification of targets for the RNA-targeting *Sulfolobus solfataricus* type III CRISPR/Cas system**

The *S. solfataricus* P2 CRISPR/Cas system has been well characterised and recently the structure of the type III-B ribonucleoprotein Cmr complex was published (Zhang, et al., 2012). This study also demonstrated that crRNAs derived from all 6 CRISPR arrays are detected in the Cmr complex, which targets RNAs complementary to the crRNA spacer sequences. CRISPRdb lists a total of 255 spacers from 7 detected arrays, which belong to the CRISPR-7 (and possibly CRISPR-11) (Kunin, et al., 2007) families. Putative protospacers were discovered using CRISPRTarget with the default settings and all predicted *S. solfataricus* CRISPRs as input (Supplement28). Of the 254 unique spacers used, 517 hits were detected for 57 spacers from 5 of the 7 arrays (Supplement29; 471 hits when E-value lowered to 0.1). An earlier study identified the targets of 29 spacers (Lilstel, et al., 2006). The top hit was a perfect match from spacer 28 in locus A (Zhang, et al., 2012) (NC_002754_3_28 in output) to an *Acidianus* two-tailed virus (AJ888457). The majority of top hits are to *Sulfolobus*, *Stygiolobus* and *Acidanus* viral sequences, but there are examples of plasmid matches (e.g. *Sulfolobus* pNOB8). One spacer in locus B (Zhang, et al., 2012)
(spacer 23 from leader end; NC_002754_4_73 in output) accounts for 393 hits, due to a very A-rich sequence. Since for Cmr no PAM has been identified and self-DNA cannot be targeted (as this system targets RNA), penalising flanking matches or searching for PAMs was not required. If analysing type III-A, rather than B, systems, mismatches between the 5' crRNA handle and the 3' flank of the protospacer DNA are important for interference (Marraffini and Sontheimer, 2010) and can be scored appropriately. However, in either case the ability to view the pairing between the handle and protospacer flanks allows matches to different CRISPR arrays to be easily distinguished.

The *P. atrosepticum* type I-F system targets a prophage in *Pectobacterium carotovorum*

Members of the genus *Pectobacterium* are economically important phytopathogens that cause a range of plant diseases (Toth and Birch, 2005). CRISPR/Cas systems in plant pathogens have not been well examined to date (reviewed in (Frampton, et al., 2012)). Previously, we analysed the type I-F system of *P. atrosepticum* SCRI1043 (previously known as *Erwinia carotovora* subsp. *atrosepticum*) (Przybilski, et al., 2011; Richter, et al., 2012), which causes soft-rot and blackleg disease in potato (Bell, et al., 2004). The *cas* genes and CRISPRs are transcribed and crRNAs generated by the Cas6f endoribonuclease (Przybilski, et al., 2011). Furthermore, the *P. atrosepticum* Csy1, Csy2, Csy3 and Cas6f proteins form a complex, which interacts with the Cas2-Cas3 nuclease (Richter, et al., 2012). Cas1 and the Cas2-Cas3 hybrid protein also interact, suggesting a role in acquisition (Fineran and Charpentier, 2012; Richter, et al., 2012). The *P. atrosepticum* SCRI1043 type I-F system contains three CRISPR arrays with a consensus repeat belonging to CRISPR-4 type (Table 3.1) (Kunin, et al., 2007). These arrays contain 41 spacers with 28, 10 and 3 spacers present in CRISPR1, 2 and 3, respectively (Table 3.2). Our previous analyses using BLAST failed to identify potential viral targets of the 41 spacers. However, spacer 6 in CRISPR2 showed 100% identity to the *eca0560* gene in its own genome (Przybilski, et al., 2011). To test CRISPRTarget, we searched for potential targets of all spacers. CRISPRFinder output files for each array were searched against ACLAME, GenBank-Environmental, GenBank-Phage, RefSeq-Microbial, RefSeq-Plasmid, RefSeq-Viral and a subset of the CAMERA metagenomic databases in CRISPRTarget (default settings, except -1/1 match/mismatch scores to penalise self matches with the 8 nt handles).
Table 3.2 Predicted CRISPR arrays in *Pectobacterium* species

<table>
<thead>
<tr>
<th>Name§</th>
<th>Type</th>
<th><em>P. atrosepticum</em></th>
<th><em>P. carotovorum</em> subsp. <em>carotovorum</em></th>
<th><em>P. wasabiae</em></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SCRI1043 (NC_004547)</td>
<td>PCC21 (NC_018525)</td>
<td>(NC_013421)</td>
</tr>
<tr>
<td>CRISPR1</td>
<td>I-F</td>
<td>28</td>
<td>38</td>
<td>17</td>
</tr>
<tr>
<td>CRISPR2</td>
<td>I-F</td>
<td>10</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>CRISPR3</td>
<td>I-F</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CRISPR4</td>
<td>I-E</td>
<td>14 *</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>CRISPR5</td>
<td>I-E</td>
<td>14 *</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

§ Names do not indicate CRISPR relationship between strains

* Likely to be one array of 29 spacers, with a 76 base spacer in the middle

The CRISPR1 array identified by CRISPRFinder was in the incorrect orientation, so CRISPRTarget was adjusted for a reverse complemented output (e.g. see Fig. 3.4). CRISPRTarget gave 67 hits from 13/28 spacers from CRISPR1 (Supplement\(^{30}\)), compared with only 2 hits when CRISPRFinder was utilised. Selection of the I-F PAM in CRISPRTarget enabled visualisation and scoring of targets that contained a consensus CRISPR-4/I-F PAM (Mojica, et al., 2009). Furthermore, the site of crRNA processing by Cas6F in type I-F systems is known (Haurwitz, et al., 2010), so 8 nt of the 5' (handle) and 20 nt of the 3' flanking regions were displayed for the crRNAs. By scoring flanks with penalties (e.g. -1/1 match/mismatch), self-targets can be penalised and moved down the output list. Usually the default cut-off score of 20 eliminates the self-matching results when default 8 nt handles are used (with -1/1 match/mismatch scores), while allowing bona fide targets. Using the same databases and increasing the E-value to 10, increased the number of hits to 406, which resulted in the identification of putative targets for 19 of the 28 spacers. A search with CRISPR1 against the GenBank-nt database with the same settings identified 21 hits for 8 spacers when an E-value of 1 was used. When the E-value was increased to 10, 24 spacers gave 85 hits scoring 20 or more, but there were some false positive sequences (eukaryotic).

\(^{30}\) DVD: Chapter3/Supplements/file4.xls
CRISPR1_19 matched a putative phage gene in _Pectobacterium carotovorum_ subsp. _carotovorum_ PCC21. Note that we denote spacer 1 as the leader-proximal spacer, but the spacer numbers in the CRISPRTarget output are numbered according to the input file. For example, since CRISPR1 was reversed in the output, spacer 19 of 28 (relative to the leader) is numbered spacer 10 in the CRISPRFinder input file. Comparing _P. carotovorum_ PCC21 (Park, et al., 2012) and _P. atrosepticum_ SCR1043 (Abbott, et al., 2005; Bell, et al., 2004) revealed that the spacer 19 target is within a 45 kb prophage containing 54 predicted coding sequences (here designated φPCC21_1; Fig. 3.5A & 3.5B). φPCC21_1 is inserted in _ryeAB_, but is absent in _P. atrosepticum_ SCR1043.

The _ryeAB_ genes are two overlapping small non-coding RNAs. In _Salmonella_ this locus is an important insertion site for prophages that have influenced this pathogen’s evolution (Balbontin, et al., 2008). Interestingly, CRISPR1 spacer 2 also matched φPCC21_1, albeit ~32 kb from the spacer 19 target (Fig. 3.5A). Mismatches in the predicted RNA–DNA hybrid, suggest that these spacers might no longer target this particular prophage, but it is also possible that they derived from a related phage. We propose that _P. atrosepticum_ has been exposed to this, or a related, phage in the past, but lysogenization has been inhibited by CRISPR/Cas.

The remaining spacers had matches to a variety of phage, prophage, microbial genome and metagenome samples (Supplement31). For example, a protospacer target for spacer 11 was identified in _Salmonella enterica_ epsilon 15 serotype-converting phage (Kropinski, et al., 2007).

---

31 DVD: Chapter3/Supplements/file4.xls
Figure 3.5 Pectobacterium prophages are targeted by CRISPR/Cas. (A) Prophage \( \Phi^{PCC21}_1 \) is targeted by spacers in \( P.\ atrosepticum \). (B) \( P.\ atrosepticum \) SCRRI1043 (top, 2761697-2811697) compared with \( \Phi^{PCC21}_1 \) in \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21 (bottom, phage co-ordinates: PCC21_018470-019020 from 2092807-2135244. PCC21 is reversed for clarity). (C) Prophage \( \Phi^{ECA29} \) is targeted by spacers in \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21. (D) \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21 (top, PCC21_017190-017500 from 1936500-1976500. PCC21 is reversed) compared with \( \Phi^{ECA29} \) (HAI9) in \( P.\ atrosepticum \) SCRRI1043 (bottom, ECA2598-ECA2637 from 2935264-2966783). (E) Prophage \( \Phi^{PC1}_1 \) is targeted by a spacer in \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21. (F) \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21 (top, PCC21_027150-027460 from 3058299-3095299) compared with \( \Phi^{PC1}_1 \) in \( P.\ carotovorum \) subsp. \( carotovorum \) PC1 (bottom, PC1_2622-2666 from 2989228-3022511). (G) Prophage \( \Phi^{PC1}_1 \) is targeted by spacers in \( P.\ wasabiae \). (H) \( P.\ wasabiae \) WPP163 (top, 2291600-2341600) compared with \( \Phi^{PC1}_1 \) in \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21 (bottom, phage co-ordinates: PCC21_018470-019020 from 2092807-2135244). (I) Prophage \( \Phi^{PC1}_2 \) is targeted by spacers in \( P.\ wasabiae \). (J) \( P.\ wasabiae \) WPP163 (top, 1192372-1236372) compared with \( \Phi^{PC1}_2 \) in \( P.\ carotovorum \) subsp. \( carotovorum \) PC1 (bottom, phage co-ordinates: PC1_3152-3199 from 3573374-3608557. PC1 is reversed). Prophages (K) \( \Phi^{ECA29} \) and (L) \( \Phi^{PC1}_2 \) are targeted by \( P.\ wasabiae \) spacers. Genome comparisons were generated using Easyfig (Sullivan, et al., 2011); genes are cyan arrows, putative prophage regions are purple and spacer target locations indicated with asterisks. Homologous regions by BLASTn are shown in shades of grey.

Pectobacterium carotovorum crRNAs match prophages in P. atrosepticum and P. carotovorum

As \( P.\ atrosepticum \) spacers matched a prophage in a related strain, we examined CRISPR targets in other representative Pectobacterium genomes. First, we uploaded the genome of \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21 (Park, et al., 2012) into CRISPRFinder and identified 5 arrays; three CRISPR-4/type I-F arrays containing 38, 3 and 3 spacers and two CRISPR-2/type I-E arrays with 14 spacers each (output in Supplement32).

Two spacers in CRISPR1 (type I-F with 38 spacers) matched different regions of eca2627 in the \( P.\ atrosepticum \) SCRRI1043 \( \Phi^{ECA29} \) prophage (Evans, et al., 2010) (also termed HAI9 (Bell, et al., 2004); Fig. 3.5C). Comparison of \( P.\ carotovorum \) subsp. \( carotovorum \) PCC21 and \( P.\ atrosepticum \) SCRRI1043 demonstrated the absence of a

32 DVD: Chapter3/Supplements/file4.xls
ϕECA29 prophage in PCC21 (Fig. 3.5D). Spacer 34 also matched a putative prophage (here designated ϕPC1_1) in P. carotovorum subsp. carotovorum PC1 (Fig. 3.5E & 3.5F). The two type I-E arrays are separated by 76 bp, so it is possible that these are one large array with 29 spacers. Spacer 8 within CRISPR4 was self-matching to its own ϕPCC21_1 prophage, but this will be non-targeting due to a position 2 seed mutation (Semenova, et al., 2011). Spacer 3 in CRISPR4 matches a transposase gene in Pectobacterium wasabiae WPP163 (Pecwa_0911), which is not predicted to be part of an island (Langille and Brinkman, 2009).

**P. wasabiae CRISPRs have targets against multiple prophages**

Next, the CRISPRs of P. wasabiae WPP163 were analysed (Supplement33). P. wasabiae has 4 CRISPRs, 2 CRISPR-4/type I-F with 17 and 25 spacers and 2 CRISPR-2/type I-E containing 16 and 6 spacers (Table 3.2). Spacers 2 and 10 from CRISPR1 (I-F array with 17 spacers) match ϕPCC21_1 (Fig. 3.5G & H), which is also targeted by the P. atrosepticum type I-F system (Fig. 3.5A & 3.5B). ϕPCC21_1 is absent in P. wasabiae, but in this location is Pecwa_2124 (a pseudogene homologous to the ϕPCC21_1 integrase) and Pecwa_2125-9. Remarkably, spacers 3, 4, 5 and 6, from the CRISPR2 (I-F array with 25 spacers), targeted genes PC1_3175, PC1_3187, PC1_3191 and PC1_3182, respectively, in a putative prophage in P. carotovorum subsp. carotovorum PC1 (here designated ϕPC1_2) that is absent in P. wasabiae (Fig. 3.5I & 3.5J). In addition, spacer 5 matches to the P2-type tail fiber protein H, eca2608, in ϕECA29 (Fig. 3.5K) and spacer 20 targeted ϕPC1_1 (Fig. 3.5F & 3.5L), which is also absent in P. wasabiae (data not shown). Therefore, P. wasabiae appears to have previously encountered phages similar to ϕPCC21_1, ϕECA29, ϕPC1_1 and ϕPC1_2, and has developed CRISPR/Cas immunity to these elements.

Overall, this analysis indicated that CRISPRTarget can reveal new targets of spacers in CRISPR arrays and demonstrates, with the example of Pectobacterium, that novel biologically relevant information can be obtained. Specifically, inter-species prophage exclusion by Pectobacterium type I CRISPR/Cas systems was suggested.

33 DVD: Chapter3/Supplements/file4.xls
3.5 Discussion

We have developed a tool designed to detect, and interactively explore, the targets of CRISPR RNA spacers. This is the first tool of this kind designed for this purpose. The inputs into CRISPRTarget are predicted CRISPR arrays or spacer sequences. These CRISPR and spacer prediction methods were initially developed in 2007-2009 (Bland, et al., 2007; Edgar, 2007; Grissa, et al., 2007) and thus do not incorporate recent refinements. These current CRISPR predictions do not take into account the direction of CRISPR transcription and errors can occur when defining spacer and repeat boundaries. CRISPRTarget enables the user to search for matches in either or both orientations of a given input and display adjacent PAM and flanking sequences. These features provide the flexibility to discover targets with PAMs and also any adjacent pairing potential, ensuring greater power in predicting biologically relevant protospacer targets.

In CRISPRTarget, the initial screen for database matches is done with BLASTn, with a range of parameters able to be defined. The default gap penalty is -10. We know of no publications that indicate that insertions/deletions are permitted in the RNA/DNA hybrid, although in some systems mismatches are tolerated (Cady, et al., 2012; Gudbergsdottir, et al., 2011; Manica, et al., 2011; Semenova, et al., 2011). The use of BLASTn allows for a smaller exact hit match of word size 7, compared to MegaBLAST (minimum word size of 28). However, BLASTn is slower (Gotea, et al., 2003). Specific databases of mobile elements (e.g. phage, plasmid, ACLAME) are provided, which reduces the execution time and increases the number of biologically relevant positives. Hits that might have high expect (E) values (e.g. >1) in larger databases will be shown as significant at the same E value in a smaller database. Not using the ‘nt’ database as the default, also avoids the showing of high scoring self-matches in the source or related genomes. Selected parts of the CAMERA databases, enriched in phage sequences, are provided (Seshadri, et al., 2007), and the user can upload custom data e.g. new genomic or metagenomic data for searching.
Following the initial BLAST screen the user can interactively refine and reduce the putative targets shown. In some systems PAMs are required, or seed sequences. These can be weighted so that only those with this feature are displayed. In the case of *S. thermophilus* DGCC7710 WT spacer 14 there is a one base mismatch to φ2972 and a base T to C substitution in the PAM. The consensus PAM for this *S. thermophilus* type II system is WT TTCTNN (or NNAGAAW on the other strand). This T was conserved in experimentally confirmed protospacers. Recent reports have demonstrated that pre-existing spacers that match to a target, but can have subtle mutations that abolish interference, increase the acquisition of new spacers in a process termed priming (Datsenko, et al., 2012; Swarts, et al., 2012). It is tempting to speculate that this spacer might increase the spacer acquisition activity of this CRISPR array against φ2972 and related phages (Barrangou, et al., 2007). The ability to detect potential targets for the type III-B system of *S. solfataricus* P2 was also demonstrated and resulted in putative targets for ~20% of the >250 spacers. Most of these were matches to archaeal viruses and plasmids, demonstrating potentially relevant crRNA targets.

To demonstrate the utility and functionality of CRISPRTarget we investigated possible protospacer targets in *Pectobacterium* species. This analysis revealed that there appears to be a history of prophage exposure and CRISPR content, indicative of an adaptive immunity against prophages. In other words, the presence of CRISPR arrays containing spacers matching prophages in other *Pectobacterium* genomes correlated with the absence of these mobile elements. The current role, if any, of these prophages is not clear. However, in the case of ΦECA29 in *P. atrosepticum* SCR1043, this prophage was shown to excise from the chromosome and circularize (Evans, et al., 2010). Furthermore, deletion of this entire prophage led to a reduction in motility and phytopathogenicity (Evans, et al., 2010), and hence CRISPR/Cas might limit the acquisition or retention of prophage-encoded virulence determinants. In our study, the detection of protospacer targets also led to the identification of new putative prophages (φPCC21_1, φPC1_1 and φPC1_2) in recently sequenced genomes. Thus, these CRISPRTarget hits enable confidence in the prediction of mobile regions of bacterial genomes, which are often poorly annotated.
*Pectobacterium* strains PCC21 and WPP163 also contained spacers that matched phage ZF40 (JQ177065), a “dwarf” Myoviridae (Comeau, et al., 2012), suggesting previous exposure to this, or a related, temperate phage. Given the phage and prophage interactions detected, it is of interest that strains WPP163, PCC21 and SCRI1043 were isolated from the USA, Korea and Scotland, respectively, over 20 years apart.

In conclusion, we have developed and tested CRISPRTarget, a flexible, interactive tool for the discovery of the targets of crRNAs in diverse databases. There is currently no comparable webserver available and thus, CRISPRTarget will provide a valuable resource for the growing CRISPR research community.
Chapter 4

Accurate computational prediction of the transcribed strand of CRISPR noncoding RNAs

This chapter is reproduced from a published paper (Biswas, et al., 2014) in *Bioinformatics* journal. Dr C M Brown acted as the primary supervisor and contributed towards the paper, especially in the introduction section. Dr P Fineran gave valuable ideas and useful comments.

4.1 Abstract

CRISPR RNAs (crRNAs) are a type of small noncoding RNA that form a key part of an acquired immune system in prokaryotes. Specific prediction methods find crRNA-encoding loci in nearly half of sequenced bacterial, and three quarters of archaeal, species. These CRISPR arrays consist of repeat elements alternating with specific spacers. Generally one strand is transcribed, producing long pre-crRNAs, which are processed to short crRNAs that base pair with invading nucleic acids to facilitate their destruction. No current software for the discovery of CRISPR loci predicts the direction of crRNA transcription. We have developed an algorithm that accurately predicts the strand of the resulting crRNAs. The method uses as input CRISPR repeat predictions. CRISPRDirection uses parameters that are calculated from the CRISPR repeat predictions and flanking sequences, which are combined by weighted voting. The prediction may utilise prior coding
sequence annotation, but this is not required. CRISPRDirection correctly predicted the orientation of 94% of a reference set of arrays.

4.2 Introduction

Many bacteria and archaea have defense systems that target incoming nucleic acids, termed CRISPR-Cas (Clustered Regularly Interspaced Short Palindromic Repeats-CRISPR-associated proteins) systems. Recent studies have shown that these RNA-mediated systems are a major antiviral mechanism in prokaryotes (Fineran and Charpentier, 2012; Makarova, et al., 2013; Sorek, et al., 2013; Wiedenheft, et al., 2012).

In bacterial genomic sequences, CRISPRs are characterised by repeated elements (repeats) interspersed with unique spacers. In most systems, RNA polymerase transcribes the CRISPR array in one direction to produce the pre-crRNA, which is processed to provide separate small RNA effectors (crRNAs) that target foreign genetic elements (Lillestol, et al., 2009; Richter, et al., 2012). The mechanism of targeting requires a number of Cas (CRISPR associated) proteins that are typically encoded nearby to the CRISPR arrays. Most CRISPR-Cas systems target DNA from bacteriophages, plasmids and mobile elements in bacterial chromosomes (Brodt, et al., 2011; Makarova, et al., 2013; Sorek, et al., 2013); however, some CRISPR-Cas systems target RNA (Hale, et al., 2009).

The direction of transcription is unknown for most CRISPR arrays and current algorithms do not predict direction. Normally, only one strand is transcribed and processed into small crRNAs, which determines base pairing target specificity. The crRNAs provide target specificity through base pairing for the CRISPR-Cas system. Therefore, target prediction algorithms permit both potential crRNAs to be analysed as the correct strand is often unknown (Biswas, et al., 2013).

There are several CRISPR discovery programs available to find these repeat arrays in genomic (Bland, et al., 2007; Edgar, 2007; Grissa, et al., 2007; Rousseau, et al., 2009) or
metagenomic sequences (Rho, et al., 2012; Skennerton, et al., 2013). The most cited of these (CRISPRFinder) locates CRISPR arrays in 46% (1080 in 2355) of the complete genomes of bacteria and in 84% (126 in 150) of archaea. As CRT and PILER-CR are better suited to automation, they are used in bacterial genome annotation pipelines (Mavromatis, et al., 2009).

Recent CRISPR-Cas classifications include three types (I-III) and at least ten subtypes (Makarova, et al., 2011). These are based on multiple criteria, including the presence of specific Cas proteins. For those type III-B systems that target single stranded RNA, appropriate assignment of the direction of CRISPR expression is essential to accurately identify RNA targets and avoid false target predictions. Whereas for type I and II, additional short specific sequences are associated 5’ or 3’ of true targets and function in different steps of the CRISPR-Cas process. These sequences are termed protospacer adjacent motifs (PAM) (Mojica et al. 2009), and recent work has suggested that these can be further discriminated into target interference motifs (TIM) and spacer acquisition motifs (SAM) that function at the different stages (Shah et al. 2013). The identification of these additional determinants requires knowledge of the transcribed strand (Biswas, et al., 2013; Shah, et al., 2013). The ability to accurately identify the target strand would also be useful to determine if during conjugation the single strand initially transferred into the recipient is preferentially targeted (Westra, et al., 2013).

Repeats may be dissimilar between the same types in different species, and the relationship between sequence and type is complex, for example near identical repeats in the same species can be associated with different types e.g. I-B and III-B (Nickel, et al., 2013) and multiple types commonly exist within a species (Makarova, et al., 2011). However, repeats can independently be clustered and grouped into sequence or structural classes (Kunin, et al., 2007). The most recent analysis puts 3527 repeats into 33 structural motif classes, 40 sequence family classes and six superclasses. Some of these classes are preferentially associated with specific CRISPR subtypes (Nickel, et al., 2013).
In some types (I and II) the repeat RNA has a stable stem loop, typically with 4-6 bases in the stem (Nickel, et al., 2013; Scholz, et al., 2013). Pre-crRNA processing occurs within the repeat, resulting in 5’ and 3’ handles derived from the repeats and attached to the spacer. This repeat structure within the pre-crRNA is required for processing in type I-F systems (Sternberg, et al., 2012). In some systems the 5’ crRNA handle is UAAGAAA derived from the 3’ end of the repeat (Maier, et al., 2013; Wang, et al., 2011). In addition, it has recently been shown that for a few species using type II systems, that crRNA transcription can occur from multiple promoter sequences (TANAAT -10 like) within the 3’ end of the repeat (Zhang, et al., 2013).

CRISPR arrays expand by acquiring specific sequences from invading nucleic acids. Although it is not completely understood, this adaptation typically involves addition of new spacers at the 5’ end of the array (Fineran and Charpentier, 2012; He and Deem, 2010; Lillestol, et al., 2009; Westra and Brouns, 2012). However, there is evidence of some systems acquiring new spacers at positions internal to the CRISPR array (Erdmann and Garrett, 2012). Spacers and repeats may subsequently degenerate, then be lost by successive point mutations, deletions, or recombination (Gudbergsdottir, et al., 2011; He and Deem, 2010). However, as clear targets of most crRNAs are not present in sequence databases it is difficult to determine if mutations have occurred in spacers. In addition, it is likely that both target nucleic acids and host crRNAs are accumulating mutations after the initial contact and spacer acquisition (Levin, et al., 2013; Sun, et al., 2013; Weinberger, et al., 2012). For example, phages can escape CRISPR-Cas interference via the acquisition of point mutations in their target sequences, and old spacers might accumulate mutations when there is no longer a selective advantage (i.e. in the absence of the target phage/plasmid) (Levin, et al., 2013; Vercoe, et al., 2013).

The 'leader' is the sequence that precedes the array and contains both the promoter and 5’ region prior to the first repeat (Pougach, et al., 2010; Przybilski, et al., 2011). In some characterised systems there is clear accumulation of point mutations in the repeats (and spacers) at the 3’ end of the array (leader-distal end). Consistent with this observation, current data indicate that new spacers are acquired at the 5’ (leader) end of the CRISPR
array and older spacers and repeats decay at the 3’ end (trailer end spacers) (Swarts, et al., 2012; Weinberger, et al., 2012; Yosef, et al., 2012).

Previous studies have predicted the direction of repeats as parts of larger studies. It has been observed that repeats in archaea are A-rich and this has been used to determine direction in that domain (Chan, et al., 2012; Shah and Garrett, 2011). Very recently Lange et al., 2014, used A-richness, and the presence of ATTGAAA(C/G) at the 3’ end of repeats, to determine the direction of repeats (Nickel, et al., 2013).

In this study we aimed to use information within, and nearby, predicted CRISPR arrays to determine the direction of transcription. Prediction was performed as much as possible independently of the use of biological knowledge (e.g. repeat class), as this is not known for many sequences. The algorithm is described and available as a Perl module for a gene prediction and annotation pipeline. This CRISPREDirection algorithm predicts with >90% accuracy the direction of transcription, and thus the correct sequences of the crRNAs that target foreign nucleic acids. CRISPREDirection will be useful for more accurate prediction of CRISPR arrays, and subsequent prediction of their types and targets, such as via CRISPRTarget (Biswas, et al., 2013).

4.3 Methods

4.3.1 Preprocessing of arrays

4.3.1.1 Prediction of arrays

Arrays were predicted with CRISPRFinder, PILER-CR and CRT on complete bacterial (2341 species with 4458 sequences) and archaeal (153 species with 236 sequences) genomes using the default parameters (Bland, et al., 2007; Edgar, 2007; Grissa, et al., 2007). Only CRISPR arrays with 2 or more spacers (i.e. 3 or more repeats) were used in the
analyses. The three programs predicted 4053, 3471, and 3166 CRISPRs, respectively. There was partial or full overlap between all three predictions for 3115 arrays. Because arrays predicted with CRISPRFinder do not show variation in individual repeats in the final output, and its dependencies are not freely available, the CRISPRFinder predicted arrays were not used. Because similar arrays were often predicted by CRT and PILER-CR, only the longer ones were used from the combined list of predictions. For overlapping predictions, the longest were selected. The final list included a total of 3571 CRISPR arrays. Those pipelines that include CRISPR annotation use a similar methodology.

Figure 4.1 Overview of the methods used in CRISPRDirection. Arrays were predicted using CRT and PILER-CR. An array with n repeats R1-Rn is shown. Parameters were calculated from the repeats or utilising flanking regions e.g. the distance to the next annotated protein coding sequence (arrows) (Methods). Relative AT% in the flanks was determined in fixed windows. For each array, a representative repeat was determined then assessed for motifs, composition, and secondary structure (Methods).

---

34 A comparable figure can be found in chapter 5 (Fig. 5.4).
4.3.1.2 Identifying representative repeat sequences

Since the PILER-CR output format provides most of the information required by CRISPRDirection, we converted the CRT predicted arrays into PILER-CR format. However, as CRT does not provide a representative repeat, we had to determine this for the arrays predicted by CRT. To find the representative/centroid repeat for each array, we used a match scoring method, then the frequency of individual repeats. We first collected all the repeat sequences of an array and the number of individual occurrences in the array. At this stage if two repeats have single base difference, they were treated as two separate sequences. We initially selected the sequence with the least mismatches as the representative repeat. Where two or more sequences had the same degeneracy score, the most common repeat was selected. If these were equal, the repeat with fewer mismatches at the ends of the array was selected. The representative repeats were collected from the 3571 arrays, and after identical or reverse complemented repeats were removed, there were 2023 unique repeats.

4.3.1.3 Reference set of arrays and repeat sequences

To develop a reference set of CRISPRs with known orientation, a set of 26 repeats with experimentally determined directions were used, 13 from Biswas et al., 2013, and supplemented with 13 repeats from the recent literature (Appendix B1). These were chosen to cover each of the ten subtypes, and a range of structural and sequence classes (Discussion and Appendix B3) (Kunin, et al., 2007; Nickel, et al., 2013). There were 208 arrays containing these exact 26 repeats.

To generate a larger reference set of arrays for our analysis, we searched the 2023 repeats for sequences similar to the 26 repeats of known orientation. In this comparison, repeats with a maximum of 3 mismatching bases were retained. This gave us a set of 120 unique repeats that were present in 575 distinct arrays. We removed arrays with >95% sequence identity using CD-HIT-EST (Supplement35 Table 2). This reduced the number of

35 DVD: Chapter4/Supplements/file1.docx
arrays to 460 arrays, 340 from 205 bacterial and 120 from 43 archaeal species. These 460 were used as a reference set in our analyses.

### 4.3.2 Calculation of Parameters

We derived and utilised a number of parameters and attributes from the repeats and regions immediately flanking the array. Each of these sets of parameters were used as a method to predict the direction of the array as either F (Forward) or R (Reverse). For some methods all 460 arrays could be used, for others not all were suitable for the analysis.

If the array was not valid for the analysis, or the analysis results in a draw, it reports NP (No Prediction). For example, if there are no mutations in an array, the reported direction by the method dependent on these will be NP (Appendix B.2). The mean PPV (Positive Predictive Value) score was calculated from 100 tests each from 100 randomly selected CRISPR arrays using resampling by re-substitution, as n (i.e. sample size) is much larger than the number of predictors to be estimated. An alternative method of resampling by 10-fold cross validation is provided in supplement\(^3\), which produced similar predictors and weights +/- 0.03. As the random probability of getting a true prediction is 0.50, we used as a score for each method its PPV minus 0.50. Scores of the predictions from each of these methods are summed, and a final prediction for each array is made (Appendix B2).

### 4.3.2.1 Screen for ATTGAAA-like sequences

The sequence ATTGAAA and variants shown in Table 4.1 were searched for on both strands of the representative repeat (Fig. 4.1). Any single base, or no base, was allowed following the motif, to allow for inaccuracy in prediction of the repeat/spacer transition in tuning the algorithm.

\(^3\) DVD: Chapter4/Supplements/file2.xls
4.3.2.2 Biased nucleotide composition in the 5' and 3' flanks- relative AT richness

In testing, we compared the percentage of nucleotides A and T (AT%) in different length windows ranging from 15 to 165 nucleotides with a step of 15 nucleotides flanking the 5' and 3' ends of the arrays (Fig. 4.1). The AT% in each of these regions were compared, in testing, to determine if they have an absolute minimum percentage difference ranging from 0.5% to 14.5%. When the AT differences for windows were greater than the minimum cutoff, a prediction was obtained. For example, if the 5' end of the array showed a higher AT% for a window size (e.g. 135) and minimum cutoff (e.g. 2.5%) the prediction was F. Similarly, the predicted direction will be R when the 3' end of the array shows a higher AT% than the 5' end of the array. For all other cases NPs are reported.

Table 4.3 Presence of ATTGAAA- like motifs at the 3' end of the array

<table>
<thead>
<tr>
<th>Sequence</th>
<th>TP</th>
<th>FP</th>
<th>PPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTGAAA(C/G)</td>
<td>130</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>ATTGAAA(N)</td>
<td>137</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>NTTGAAA(N)</td>
<td>140</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>ANTGAAA(N)</td>
<td>140</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>ATNGAAA(N)</td>
<td>148</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>ATTNAAA(N)</td>
<td>143</td>
<td>18</td>
<td>0.88</td>
</tr>
<tr>
<td>ATTGNAA(N)</td>
<td>111</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>ATTGANA(N)</td>
<td>137</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>ATTTGAAN(N)</td>
<td>141</td>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>WWWGAAA(N)</td>
<td>240</td>
<td>9</td>
<td>0.96</td>
</tr>
</tbody>
</table>

(N): any or no base, W: A or T, TP: True Positive, FP: False Positive, PPV: Positive Predictive value
4.3.2.3 Degeneracy at the 3' end of the array

The array was divided into three sections: 5', middle and 3' sections, each with an equal number of repeats, where possible, or the nearest integer. For arrays with 3 or 4 repeats, both 5' and 3' sections had one repeat each. Of the 460 arrays, 164 had no mutations in the array, and were excluded from this analysis. The remaining 296 had one or more mutations and a direction could be predicted. We calculated the total number of mutations present in these sections. In this analysis, a base was scored as mutated only when it differed from an adjacent repeat. This rule of mutation was necessary as many long CRISPR arrays were shown to have a point mutation, which is propagated to the next repeats, resulting in two common repeats in a single array. Often, both the repeats were found to have over one third of repeats associated with them, making it difficult to select one as the representative repeat. The score for the 5' and 3' sections were calculated in the following manner:

\[
\begin{align*}
5' \text{ section score} &= \text{Total number of mutations} + \text{Observed mutation in the last repeat.} \\
3' \text{ Section score} &= \text{Total number of mutations} + \text{Observed mutation in the last repeat.}
\end{align*}
\]

The ‘observed mutation in the first/last repeat’ can either be True (1) or False (0), this was useful to correctly identify the direction when the total number of mutations in both 5' and 3' section was equal. The end with the lowest score was predicted to be the 5’ end.

4.3.2.4 Potential for RNA secondary structure in the repeat

For this analysis we first removed the redundant repeats from the sample (Supplement Table 2) and used RNAfold to measure the Minimum Free Energy (MFE) of the repeats in both directions (Hofacker, et al., 1994). In addition, we masked bases from 0 to 11 bases from both ends, and calculated the MFE of the center part of the repeats and their reverse complements. The direction with the lowest free energy for those with differences of >0.5 kcal/mol was scored as forward.

37 DVD: Chapter4/Supplements/file1.docx
4.3.2.5 A relative to T in the repeat- A/T ratio

For this analysis, the relative number of A to T in the representative repeat was measured and the direction with the highest A/T ratio scored as forward (Discussion 4.5.2).

4.3.2.6 Distance to the next coding gene at the 5' and 3' end- length of leader analysis

The positions of next CoDing Sequences (CDS) were obtained from their annotation files (NCBI Genbank format, gbk) using the start and stop positions of the 460 arrays. All CDSs overlapping any CRISPR arrays were omitted (99 CDS). The closest CDS coordinate on the 5' end of the array and the closest CDS coordinate on the 3' end of the array were analysed. The end with the longest distance was scored as forward.

4.3.2.7 Parameters used by CRISPRDirection.pl.

Method (from sections 4.3.2.1-4.3.2.6): parameters if any, weighted score (PPV-0.5). The high precision parameters and scores are in parentheses. A flow diagram of the methods is shown in Appendix Fig. B1.

(i) Specific ATTGAAA(N) motif: score 0.50 (4.5)
(ii) Biased AT richness in flanks: window 60 (120), AT% difference >= 10 (14), score 0.32 (0.45)
(iii) Degeneracy at 3' end: minimum mutations required 1 (2), score 0.41 (0.46)
(iv) RNA secondary structure: bases ignored at both ends 5 (8), score 0.37 (0.39)
(v) Longer distance to next CDS: ratio >200%, min length of longer sequence 75, score 0.18 (0.18)
(vi) A/T ratio in repeat: score 0.37 (0.37).
4.3.2.8 Final prediction.

The maximum score possible for forward or reverse is the weighted sum of individual scores of the methods (e.g. Total_Forward_score, Appendix B.2). NP were not counted. The Score_difference = absolute value of (Total_Forward_score - Total.Reverse_score). High confidence: the score difference is >=66% of the sum of the scores and >0.5; Medium: >=33% and <66%; Low: <33%. The minimum score difference for "High" confidence is also required, as this excludes results from less accurate prediction methods, where the more sensitive methods did not score (NP). The output is shown in Supplement38 Table 4, and the flow diagram of the algorithm is shown in Appendix B.2.

4.3.2.9 Availability.

An implementation in Perl with documentation and test data is available at http://bioanalysis.otago.ac.nz/CRISPRDirection. This will take as input- a fasta or Genbank formatted, annotated or unannotated genomic sequences, or simply repeat sequences in fasta format.

4.4 Results

We aimed to develop an algorithm to detect the direction of CRISPR transcription. This would, as much as possible, be independent of specific biological knowledge for known types of arrays, as often this is neither known nor predicted. Such a method could be applied within a pipeline to new instances and potentially new classes of arrays. The predictive methods adopted were i) searching for ATTGAAA-like sequence motifs in the repeats, ii) finding biased nucleotide composition (relative AT richness) in the leader and trailer regions of the arrays, iii) analysis of mutation(s) in the repeats (array degeneracy), iv) analysis of RNA secondary structure in the repeats, v) finding the longer distance to the

---

38 DVD: Chapter4/Supplements/file1.docx
next coding gene at the 5’ and 3’ ends and (vi) greater number A compared to T in the repeat. These methods are depicted in Fig. 4.1.

A reference set of known direction was assembled by using a set of 26 repeats of experimentally determined orientation, in 135 unique arrays, This was extend to 460 repeats by allowing up to 3 mismatches (Methods 4.3.1).

### 4.4.1 Specific sequences at the 3' end of the repeat

A recent analysis of repeats noted that about a third of their set of repeats end in ATTGAAA(C/G) and they used that to predict repeat direction (Nickel, et al., 2013). To investigate if this was a useful predictor on our independently created reference set we searched for variants of ATTGAAA(N) at the 3’ end on both strands. An exact match was found in 30% (137) of the 460 reference arrays. In all these cases this correctly predicted the direction of the array (PPV=1.00, Table 4.1). Indeed, similar one base variants of all the bases except the G slightly increased the number of predictions but retained PPVs of 1.00 (Table 4.1). For those 137 arrays that had ATTGAAA(N) we used this as a certain predictor of the direction of these arrays. This parameter was given a score of 0.50 (PPV of 1.00-0.50 random chance).

### 4.4.2 Biased AT composition in the 5' and 3' flanks.

The promoter/leader sequence 5' of the CRISPR array would be expected to differ from the terminator sequence following it. These promoters have been described as being AT rich in several studies, and some AT rich sequences (e.g. TATAAT like sigma 70 promoter -10 elements) have been shown to be functional in CRISPR promoters (Lillestol, et al., 2009; Pul, et al., 2010). The AT% was calculated in windows at the two flanks and the % difference compared. An AT% difference of 0.5 to 14% and windows up to 165 bp long were compared (Methods 4.3.2.2). The analysis was repeated 100 times for a new randomly selected set of 100 CRISPR arrays in each run, and the results were compared to determine
the window length and minimum percentage difference in the AT that gave an optimal PPV (Appendix Fig. B4). This sampling approach was used to calculate measures of variability within our set of 460, which is a sample of all possible CRISPR arrays.

Figure 4.2 Relative AT richness in the two flanks. AT% for windows of the specified lengths were calculated on either flank, the side with the higher AT% was used as a predictor of the leader. TP: True positives, FP: False positives, NP: No Prediction, PPV: Positive Predictive Value. Data are the mean and standard deviation of 100 samples of 100 from the reference set of 460 arrays.

As shown in Fig. 4.2, a window size of 60 with a minimum AT% difference of 10% gave a high PPV (0.82) with a tolerable NP (TP 39, FP 9 and NP 53). Higher PPVs could be achieved, e.g. window size of 105, AT% difference >14%, PPV 0.95 (Fig. S1). However, this combination could only be applied to one third of the data. As this method is one of a combination to predict the probable direction, we used a window size of 60, >10% AT% as a balanced default for CRISPRDirection. Alternative combinations can be applied in the implementation and results are shown in Fig. S1.

4.4.3 Degeneracy at the 3' end of the array

Based on the idea that repeat sequences at the 3' end may have accumulated mutations, we looked for differences at either end of the predicted arrays. Fig. 4.3A shows that a PPV
score of >0.90 could be achieved (TP 89, FP 9 and NP 2). Some of the FPs may be caused by a random mutation, or rare sequencing errors within the arrays. Indeed, a higher accuracy (PPV 0.95) can be achieved by only including arrays with a minimum of two or three mutations (Fig. 4.3A). However, as that increases the number of NP (to ~40%) we did not set any minimum mutation requirement by default. For 296 valid arrays (Methods 4.3.2.3) this method had 263 TPs, 28 FPs and 5 NPs.

4.4.4 Potential for RNA secondary structure in the repeat

Some CRISPR repeats are known to form small RNA secondary structures, which would be strand-specific. We used a simple method to test for the greater ability to form thermodynamically stable secondary structures on each strand. This method was designed to avoid requiring prior knowledge of classes, or structures in different classes when predicting the direction.

The hypothesis was that the minimum free energy (MFE) of the forward strand would be lower compared to the MFE of the other strand. In an initial analysis we observed that calculating the MFE of the whole repeats gives a poor prediction accuracy (Fig. 4.3B, PPV 0.55). The RNA secondary structural elements, where experimentally characterised, are typically present towards the center of the repeats rather than at the ends (Nickel, et al., 2013). Therefore, we progressively masked the leading bases from the both ends of the repeats and calculated the MFE (Fig. 4.3B).
Figure 4.3 Repeat parameters. A). Mutations in the repeats at the 3’ (leader-distal) end of the array (Section 4.4.3); B). Relative stability of predicted RNA secondary structure (Section 4.4.4); C). Relative A/T ratio (Section 4.4.5). TP: True positives, FP: False positives, NP: No Prediction, PPV: Positive Predictive Value. Data are the mean and standard deviation of 100 random samples of 100.
The accuracy of the prediction increased as the ends were masked by up to 5 bases (Fig. 4.3B. PPV 0.87). Further base masking showed that although the PPV was increased the number of true predictions rapidly reduced (e.g. PPV 0.90 NP 26% at 8). The average repeat length is 31 nucleotides, removing up to 5 bases from both sides (total 10 bases) still kept the secondary structural elements intact in the middle for most cases, and gave a more accurate prediction. However, as more bases were masked, only longer repeats gave useful results, indicated by the rapid drop off after 8 bases in Fig. 4.3B.

### 4.4.5 Analysis of A relative to T in repeats – A/T ratio analysis

Several studies have also used A/T ratios in the repeat as a predictor of direction, as it has been observed that archaeal repeats are A rich (Chan, et al., 2012; Nickel, et al., 2013). The notion is the forward strand will have more A than T. The PPV for this was 0.72 (68 TP, 27 FP, 5 NP of samples of 100). During this analysis we observed that AT rich repeats had a higher rate of false predictions. To investigate if this has a clear pattern, which can be applied to improve the overall prediction, we separated the repeats into two groups. The first group contains the AT rich repeats (i.e. AT% > GC%) and the second group contains AT poor repeats and we reapplied the method. The result (Fig. 4.3C) shows that this method is better applied to repeats that are not AT rich (PPV 0.87).

### 4.4.6 Distance to the next coding gene at the 5' and 3' end- length of leader analysis

The ‘leader’ region containing the promoter may be longer than the ‘trailing’ region containing the transcription terminator (Fig. 4.1). To test if this could be used to predict the direction, the positions of the next coding gene (CDS) at the 5' and 3' end were obtained from their corresponding Genbank annotation. Only this method, of the prediction methods implemented, relies on other types of gene annotation. The distances were compared to obtain a parameter that could be tested for utility in prediction. As yet the NCBI Reference sequences do not include CRISPR predictions and leave them as long intergenic gaps.
Hence, CDS annotations may not be accurate in these regions and it is possible to have undetected CDS close to the CRISPR arrays, or falsely predicted CDS close or within the arrays. Indeed 99 predicted probable arrays showed overlapping predicted CDS. These overlapping CDS were ignored for this analysis.

![Figure 4.4 The distance to the next coding sequence was determined for each flank.](image-url)

The distance to the next coding sequence was determined for each flank. The side that was longer was predicted to be the leader. TP: True positives, FP: False positives, NP: No Prediction, PPV: Positive Predictive Value. Data are the mean and standard deviation of 100 random samples of 100. PPV values are poor (near random) when the lengths are similar (~1). The highest PPV (0.68) is obtained when the ratio is >200%.

Distances ranged from 2 to 5252 bases to the next CDS for the set. However, often both distances were quite short (<75) making prediction unreliable. Hence, instead of only testing for the longer sequence, we also compared if the longer sequence was at least 120%, 140, 160, 180, or >200% the size of the shorter, and required that at least one of the lengths be over 75 nucleotides (Fig. S4). No Prediction (NP) was made if the ratio did not meet the criteria or the length was too short (Fig. 4.4). These predictions can be found in the supplementary spreadsheet\(^{39}\). The maximum PPV obtained was 0.68 (>200%, Fig. 4.4), this was selected to be used by default. However, about half the arrays did not have such long leaders and resulted in NP calls (Fig. 4.4).

\(^{39}\) DVD: Chapter4/Supplements/file2.xls
4.4.7 Combined Prediction

Each of the six predictors were combined and weighted according to their PPV. In order to weight more accurate predictions higher (i.e. assigning higher score to more accurate methods), each of the six predictions were weighted according to PPV. Thus, the highest weighted prediction was ATTGAAA(N) (0.5) followed by degeneracy (0.41) and the lowest weighted prediction was longer leader (0.18). The combined method correctly predicted 424 arrays out of the 460 CRISPRs used as reference with a PPV of 0.94, FP 27 and NP 4 (Bacteria: PPV 0.92, TP 317, FP 19, NP 4 and Archaea: PPV 0.93, TP 112, FP 8, NP 0).

To further assess the robustness of the method, the algorithm was run without each of the methods in turn and without combinations of related methods. Leaving out the most accurate predictor (ATTGAAA(N) Table 4.1) had a small effect reducing the PPV by ~4% (Appendix Fig. B3). Leaving out the degeneracy had the greatest single impact on PPV.
reducing it to 0.91. Leaving out the longer leader method, the only measure that depends on CDS annotation reduced had little effect. If only repeat sequences were used as input, an option in the implementation, PPV was 0.87 using three methods (i.e. Motif search, A/T ratio, and RNA structure analysis) that only require the repeat.

An alternative prediction strategy was to predict using the highest practical PPV from each method (window size 120 and minimum AT% difference 14 in relative AT richness analysis, 8 masked bases in the RNA secondary structure analysis, and by requiring two or more mutation in the array degeneracy analysis). Using this alternative set of options stated as ‘high precision parameters’ in the Methods section (4.3.2.7), we obtained a PPV of 0.95 against the whole reference set, with 413 TP, 21 FP and 26 NP (bacteria: PPV 0.95, NP 26 and archaea: PPV 0.97, NP 4, Supplement).

Predictions differ in total scores, therefore a prediction confidence could be determined. Predictions are flagged with 'high' confidence if they met defined criteria (4.3.2.8, Appendix B.1). Final predictions, and scores, for the direction for the full dataset of 3571 arrays are available in the supplement.

4.5 Discussion

The algorithm described works surprisingly well using a set of relatively easily calculable parameters. At the default level of accuracy it correctly predicts 94% of reference arrays. Higher levels of accuracy (e.g. PPV 0.95 and NP 26) are possible by changing some of the parameters. This method can be built into a generic prokaryotic prediction pipeline, without a requirement of knowledge whether the sequence is from bacteria or archaea and without the need to define the type, subtype, class or superclass of the array or repeat. However, it

---

40 DVD: Chapter4/Supplements/file2.xls
41 DVD: Chapter4/Supplements/all_3571_arrays.txt
is also possible for users analysing specific genomes to incorporate biological knowledge, e.g. specific motifs or presence of RNA structure into the prediction.

### 4.5.1 Reference Set

The method was tested on a set of 460 arrays, these consisted of 208 arrays having exact matches to a set of 26 experimentally verified repeats from all 10 subtypes. This set was extended by allowing a small number of mismatches (up to 3 in repeats of mean length 31). Within genera and classes of arrays there is often a large amount of substitution in arrays (Kunin, et al., 2007). Including this small tolerance might include some non-functional arrays, but is unlikely to reverse the arrays. We compared our array sets to the repeat sets used by Lange et al., (2013). Our set had members in 7 of the 10 largest structural classes and 9 of 10 largest sequence classes described (Appendix Fig. B2, Supplement Table 9) (Lange et al., 2013). This set provides a new independent reference set for CRISPR analysis.

The similarity to known repeats could be extended to provide a predictive measure; however, its validation would require a different reference set. We generated another reference set by using the 528 arrays that have ATTGAAA(N) at the 3' end, of these 140 had <4 matches to the 26 known repeats (Appendix B.1). All 140 were correctly orientated. Up to 7 mismatches provided a PPV of 1.0 (250 arrays), 43% of all the arrays had this amount of similarity. This alternative approach is also implemented in CRISPRDirection. The use of this motif as a predictor (or any user defined motif) can be included optionally in the CRISPRDirection program. The score can also be changed, for example increased to 4.5, to override the other predictions.

In both previous sets there are clearly some sequences with incorrect orientation, based on subsequent experimentally verified repeats (Kunin et al., 2007 and Lange et al., 2013). For example, the type I-F systems have a structured repeat that commonly ends in

---

42 DVD: Chapter4/Supplements/file1.docx
CUAAGAAA (Fig. 4.6) (Haurwitz, et al., 2010; Mojica, et al., 2009; Przybilski, et al., 2011; Richter, et al., 2012; Sternberg, et al., 2012). These are represented in the classification of Kunin et al. (2007) as CRISPR-4. They comprise classes, M4 and F5, and incorrectly, its reverse complement, M6 and F8, in the classification of Lange et al., (2013). The work described here will allow further merging and refinement of these repeat classes, after prior prediction of the repeat direction.

**Figure 4.6 Base conservation in repeats associated with type I-F systems.** Classes F5 and the reverse complement of F8 (Lange et al 2013) were merged to form one class, the orientation was determined experimentally from (Przybilski, et al., 2011) and correctly predicted by CRISPRDirection. There are 109 Sequences of this class in our reference dataset. The true length is 28 bases (3-30 on diagram). Extra base(s) at the left (numbered 1-2) and right (31) ends of some predicted repeats (6, or 22 of 223), in addition, some predictions are shorter. These are due to mispredictions of the ends of the repeats. The full alignment is in supplement Table 7.

### 4.5.2 Specific methods

We utilised one specific motif in this analysis (AUUGAAA) as it was present in one third of arrays. In several systems cleavage immediately before this sequence generates the 5' handle on the crRNA (Nickel, et al., 2013; Scholz, et al., 2013). Variations on this also provided strong predictors, as long as the G was retained (Table 4.1). These or other variants could also be used as predictors, particularly if the type were known. For example, to predict type I-F associated CRISPRs, a variant of this in which there is reordering of the leading AU right region, may be used (UUAGAAA) (Fig. 4.6). We searched for motifs at the end of the sequence but also allowed an additional base to be present in the repeat prediction. Current CRISPR array prediction algorithms cannot always accurately predict

---

4 DVD: Chapter4/Supplements/file1.docx
the termini of repeats, for example if the first base of the following spacer is also repeated (e.g. Fig. 4.6 and Supplement Table 7).

Additionally, users can define other repeats specific sequence motifs. For example, the presence of motif ‘GUU’ in the first bases of the repeats improved accuracy, and TATAAT like sequences function in some repeats (Zhang, et al., 2013). However, as some CRISPRs (e.g. I-E) do not obey these rules, and since our aim was to make the analysis unbiased to specific properties, we did not use more complex motif analysis by default here.

The relative AT% on either side of the repeat provided a useful predictor for 177 of the 460 arrays (Fig. 4.5). This is likely to represent the presence of AT rich promoter sequences (e.g. TATAAT) and GC rich structured terminator sequences. In addition, the leader region is important for acquisition of new spacers, which may require AT rich binding sequences (Diez-Villasenor, et al., 2013; Yosef, et al., 2012). The automated determination of the direction of arrays will enable further computational analysis of these regulatory regions (Pul, et al., 2010). The relative AT% could also possibly represent adjacent tracrRNA (Type II) or Cas genes (Zhang, et al., 2013).

For 197 of the reference arrays, the presence of a greater number of mutations at the 3’ trailer end provided a specific predictor (Fig. 4.3A). We believe this reflects the degeneration of older repeats at the trailer end. Therefore, CRISPRDirection will facilitate studies of the evolution of these arrays (Gudbergsdottir, et al., 2011; He and Deem, 2010; Weinberger, et al., 2012).

Calculation of a likelihood of forming an RNA structure worked poorly as a predictor on full length repeats (Fig. 4.3B). This was surprising as some types have structures (I, II) and these are well represented in our reference set. However, these structures are commonly small and located near the centre of the repeat, with the termini

---

44 DVD: Chapter4/Supplements/file1.docx
involved in other processes (see above). Exclusion of the five bases at each end (e.g. UGAAA in over 1/3 of the sequences) improved the PPV (Fig. 4.3B).

The relative number of A compared with T in repeats has previously been utilised to predict CRISPR direction. In particular this is used as a predictor of the directions of arrays in the UCSC Archaeal Genome Browser (Chan, et al., 2012). This gave a PPV of 0.72 overall, but was most effective (0.86) when applied to repeats with <50% AT (Fig. 4.3C).

We examined the distance to the next annotated coding sequence (CDS) as a predictor (Fig. 4.4). This was not a very effective predictor unless the sequences were relatively long. This is the only measure used here that depends on prior annotation of the genome, and its exclusion (Appendix Fig. B3) only reduced precision by 1%. Therefore, when CRISPRDirection is used as part of a pipeline it could be run prior to CDS prediction. This order would have the advantage of avoiding the many likely spurious CDS predictions within CRISPR arrays (e.g. 99 of 460 arrays in the RefSeq annotation). CRISPRDirection could be built into such pipelines, such as that used by the JGI-DOE (Mavromatis, et al., 2009). A disadvantage of CRISPR prediction prior to CDS prediction is that repeats in some proteins may be falsely predicted as arrays by CRISPR prediction algorithms.

The combined analysis is robust to the removal of any of the predictors, or combinations of predictors (Appendix Fig. B3). Normally a prediction is made that depends on several methods. This score ranges from 2.6 (where all parameters when added are concordant), to ~0.18 (when predictors are discordant or most are NP). For each prediction the combined score can be used to estimate the confidence of the score, high, medium or low. The 32, mainly bacterial arrays, for which a false prediction was made had no obvious similarity in terms of species origin, repeat sequence, or likely class (Supplement Table 6). These predictions mainly had low confidence (30/32).

In the future, CRISPRDirection could be built into a more sophisticated CRISPR array finding algorithm. This could include additional automatically calculable parameters,

---

45 DVD: Chapter4/Supplements/file1.docx
or user inputted parameters. We observed that specific sequences are common near the ends of the repeats, such as GAA 3', GTT 5' and specific sequences are also found the leader/promoter region. These could likely be used to refine the prediction algorithm, particularly in those cases where the direction is unable to be called with confidence. This could also be provided in a later interactive CRISPR prediction interface where expert users could input relevant predictors, or balance the parameters for their own species.
Chapter 5

CRISPRDetect: A flexible algorithm to define CRISPR arrays

This chapter is reproduced from the paper to be submitted to Bioinformatics journal. Dr C M Brown acted as primary supervisor. Dr P Fineran provided list of important biological features that were added to the project. He along with collaborators and lab members critically evaluated the software.

5.1 Abstract

CRISPR RNAs provide the specificity for a recently discovered noncoding RNA guided defence system in prokaryotes. CRISPR arrays consist of repeat sequences separated by specific spacer sequences. CRISPR-like structures have been detected in over 50% of bacterial and over 90% of archaeal genomes. However, currently available detection algorithms do not utilise recently discovered features of CRISPR structure and expression, nor predict the strand of RNA transcription. We have developed and implemented a new approach to automatically detecting and interactively refining CRISPR arrays. CRISPRDetect discovers putative arrays, extends the array by detecting additional repeats, and optimises the internal structure. Specific examples are described for bacterial and archaeal genomes. CRISPRDetect enables more accurate detection of arrays and is suitable for inclusion in genome annotation pipelines. It is available as an interactive web program and source code at bioanalysis.otago.ac.nz/CRISPRDetect.
5.2 Introduction

The CRISPR/Cas system is an adaptive immune system in prokaryotes. Specificity is provided by short non-coding RNAs that recognise invading foreign DNA. These non-coding RNAs consist of arrays of near identical direct repeats (DRs) typically 22-35 bases long punctuated by short non-identical 'spacer's that determine the specificity of the system (Diez-Villasenor, et al., 2013; Westra, et al., 2013). Computational recognition of potential CRISPR RNA genes in genomic DNA has been approached in different ways. Initially CRISPRs were predicted by genomic pattern matching programs such as ‘scan for matches’ (Benson, 1999; Dsouza, et al., 1997; Edgar and Myers, 2005; Guan and Zhao, 2005; Jurka, 1998). However this approach has limitations, particularly in distinguishing CRISPRs form other types of repeats in the genome. In addition most of the arrays show some mutation (substitutions or insertion and/or deletions), particularly at the 3’ end of the CRISPR arrays and more sophisticated approaches are needed to represent them properly.

A number of tools have been developed to predict CRISPRs including both command-line executable programs (e.g. CRT and PILER-CR) and web-applications (e.g. CRISPRFinder, CRISPI), facilitating CRISPR prediction and analysis (Bland, et al., 2007; Edgar, 2007; Grissa, et al., 2007; Rousseau, et al., 2009). However, a major drawback of the existing methods is that predictions do not use recent biological information. Previous methods mostly rely on base similarities (and often on length distribution) in the repeats and spacers with predefined parameters, hence fail to detect key features of CRISPRs. For example, insertion, deletion and multiple point mutations may occur and propagate through subsequent repeats, or a portion or whole repeat and/or spacer could be deleted. Many of the existing programs fail to detect short or degenerated CRISPR arrays. Setting the parameters with high sensitivity may include these but will also include many non-CRISPR genomic repeats. Finding the true positives from such a significantly large list of short CRISPR-like regions is difficult and compute-intensive. Instead of solely depending on length distribution, we propose that scoring putative arrays with CRISPR properties from verified CRISPRs will help to separate true positives from the spurious ones.
Current evidence and modelling indicate that arrays are extended by duplication of the repeats and acquisition of spacers from the invading DNA (Fineran and Charpentier, 2012). Both repeats and spacers can be lost by mutation, small and large insertion(s) or deletion(s), or recombination (Jiang, et al., 2013). Modelling has indicated there is a dynamic flux between acquisition and loss, driven by mutation and selection (Weinberger, et al., 2012).

Recent CRISPR related bioinformatic analyses have shown (e.g. CRISPRDirection, CRISPRmap) that the repeats can indicate the direction of the CRISPR transcription and family (Biswas, et al., 2013; Lange, et al., 2013). For example, conserved sequence motifs (e.g. ATTGAAA) found at the 3’ end of the repeats, can be a clear indicator of the transcriptional direction. Hence, it is important to accurately predict the repeat/spacer boundaries while predicting CRISPRs. On the other hand, spacers are used for identifying the targets (termed protospacers) of the system. However, as the spacers are quite short it may be difficult to separate the true targets from a large list of potential hits. Every additional correctly annotated base of a spacer can assist target detection. In particular the crRNA bases towards one end of the spacer are part of a ‘seed’ sequence, which is the most critical requirement for base pairing, target recognition and interference. Similarly, it is also important to correctly identify the precise ends of the spacers to enable accurate prediction of flanking motifs, termed protospacer adjacent motifs (PAMs), which are essential for effective targeting. The 3’ ends of arrays are often not detected due to the degeneracy in the repeats as well as the presence of partially deleted repeats and/or spacer sequences.

Insertions and deletions are quite frequent especially in the 3’ end of the array. Currently, PILER-CR is the only program that handles insertion(s) and/or deletion(s) in repeats, resulting in repeats with un-equal lengths. Both CRT and CRISPRFinder match repeat lengths by shifting bases into or out of the spacer compromising both the spacer and repeat sequences.

Often, some parts of the CRISPRs (generally observed towards the 3’ end) undergo severe mutations and deletions, resulting in partial or total repeat, and/or spacer deletion. The order of spacer acquisition and loss implies a particular evolutionary history. It would
be useful if the detection of CRISPRs provided a potential extension with lower repeat identity to test if degenerated, but still recognisable, repeats are present.

Current predictions specify arrays on the forward strand, thus ~50% of the arrays can be in the wrong orientation. This has recently been addressed in CRISPRDirection, included in the analysis presented here.

The type of a CRISPR is dependent on a specific set of Cas genes, and repeat sequence. Multiple arrays of different types can be found in a single genome. Although the Cas genes are often nearby the array, the distance can be large, making it difficult to classify a CRISPR array to a particular type. Existing prediction programs do not predict the type of the CRISPRs, but may provide lists of Cas genes present in the genome (Rousseau, et al., 2009) or nearby region (Grissa, et al., 2007). A recent study (Lange, et al., 2013) uses the similarity of the CRISPR repeats to a classification of clustered repeat families for identifying CRISPR types and classes.

Most CRISPRs differ in content and function, and often require a distinct set of rules or manual intervention to represent them properly. But as yet, there are no tools or web-application available, which allows a true array specific analysis.

5.3 Materials and Methods

CRISPRDetect uses extensive analysis to identify true CRISPRs. Fig. 5.1 shows a schematic of the CRISPR identification and refinement process. CRISPRDetect supports identification of CRISPR array with minimum of >=2 repeats (i.e. minimum one spacer), with a default word length of >=11. However, most valid (likely true positive) CRISPRs can be identified with the default word length (e.g. 11) and minimum repetition (e.g. 3 or above). It uses five main processes to analyse a potential CRISPR, namely i) direct repeat detection- putative CRISPR, ii) removal of tandem repeats, iii) refinement, iv) finding

---

46 DVD: Chapter5/CRISPRDetect/CRISPRDetect_v1.0.pl
direction and family and v) scoring and validation. Most of these processes use multiple sub-processes depending on the requirements and nature of the putative CRISPRs.

Figure 5.1 The CRISPRDetect automated pipeline. The modules that make up the pipeline are shown. In some cases there is iterative repeat of the processes. See CRISPRDetect.pl (attached DVD). The interactive version (bioanalysis.otago.ac.nz/CRISPRDetect) allows dynamic alteration of the parameters to suit the CRISPR type and genome.

5.3.1 Detection of direct repeats - putative CRISPRs

Putative CRISPRs are identified as two short stretches of identical sequence separated by a dissimilar short sequence. Since the minimum length of all experimental verified CRISPR repeats are above 23 nt, we used, by default, a much shorter minimum word size of 11 (can be changed to any positive integer >5) for the repeating words. The minimum and
maximum space (potential spacer length) between words are calculated using the following formulas.

\[
\text{Minimum space between repeating words} = 30 - \text{repeating\_word\_length}
\]

\[
\text{Maximum space between repeating words} = 125 + \text{repeating\_word\_length}
\]

The idea of not using fixed minimum and maximum lengths is to reduce user input as well as maintaining the speed. As, the shortest valid spacer length is well over 20 nt, this approach will ensure that CRISPRDetect will not miss any potential true CRISPR. The 11 nt word size ensures that shorter putative CRISPRs with multiple base mismatches will not be excluded, while speed is not being compromised for a typical bacterial or archaeal genome. This is done using the RegExps (Regular Expressions) in Perl.

5.3.2 Removal of tandem repeats

The genomic regions containing the putative CRISPRs are analysed to identify clear direct repeats. Using the repeating word, we split the genomic region into shorter sequence stretches (every stretch begins with the repeating word), which are then aligned using clustalW (version 2) (Larkin, et al., 2007). The ClustalW alignments are checked to increase the initial repeat length as well as to identify tandem repeats. Putative CRISPRs that identified as clear tandem repeats are then removed from the list. This typically still retains ~50 putative CRISPR per average sized (~4mb) genome (using default word size and minimum number of repetition). However, for some bacterial genomes with large number of genomic repeats, the list of putative CRISPRs can be over multiple hundreds to thousands.

5.3.3 Refinement of putative CRISPR

CRISPRDetect supports eight independent refinement subroutines. These methods are used by default and applied in the specified order. Fig. 5.2 shows the schematic diagram of the
CRISPRDetect analysis pipeline, which is detailed in the following sections. However, each of these methods can be applied independently in an interactive manner to one or all CRISPRs in the CRISPRDetect web-server.

5.3.3.1 Extend the repeat ends

Mutations at the ends of repeat may result in being included as part of the adjacent spacer sequences. CRISPRDetect progressively extends the repeat length on sides, comparing the bases from the following spacers or flanking regions with a dynamically obtained minimum percentage column identity (column refers to sets of aligned repeats followed by spacers). The percentage column identity is determined by the following formula:

\[
\text{minimum percentage column identity} = 100 - (\text{number of repeats} \times 10) - 5
\]

As for the longer (e.g. arrays with more than 9 repeats) putative CRISPRs the above score will be negative, the lower limit is set to 20. The dynamic minimum percentage identity is very crucial, as a fixed value may not always work for arrays with varying number of repeats. For example >50% column identity works well for arrays with more than 4 repeats, but shorter arrays need higher identity cutoff (e.g. >=75% for arrays with 4 repeats, >=66% for arrays with 3 repeats, 100% for arrays with 2 repeats).

In shorter arrays there might be a column with low (e.g. 2/4, 50%) identity followed by a highly identical region (potential repeat region). CRISPRDetect uses an adaptive method to handle this issue where instead of using one column identity as cutoff, it uses an additional column identity (i.e. alternate percentage column identity). The alternate column identity is half the primary column identity and only applicable when a column identity is lower than the primary column identity but followed by minimum two columns bettering the primary column identity. To identify non-CRISPR tandem repeats, this method works well to extend seed repeats to full-length tandem repeats.
5.3.3.2 Selecting representative repeats

For most arrays there is very little dissimilarity among repeats and a representative repeat is easily determined. It may be a challenge to identify a single representative repeat for shorter CRISPRs, those with frequent mutation in the repeats, or when more than one repeat is found in longer arrays.

The precise representative repeat is an important component of an array, as it helps to identify the family, direction, true spacer lengths as well as the degenerated repeats at the end of array. This method is used after every major operation on the array to identify the true representative repeat. The method used is previously described. It results in the most common repeat in the array becoming the representative, with the next most common being the 'alternative' repeat.

5.3.3.3 Extending an array

This method progressively checks the flanking regions of the CRISPR arrays within a distance equal to 1.25 times the length of the representative repeat plus the median spacer length.

The permitted lowest gap between newly found repeat and existing repeat is 0 nucleotides to address total spacer loss, and the default upper limit is 125 nucleotides or user given gap length between 2 CRISPRs. The flanking region is compared with smith-waterman algorithm (EMBOSS/water) (Rice, et al., 2000) with increasing gap-penalty (starting from 5.5 up to 10 with increase of 0.5 in each turn) to identify the best and longest non-gapped alignment. Once such an aligned region is identified, the region is extended either side accordingly to match the representative repeat length. It is then further checked to ensure that the minimum repeats identity (default >=75%) is valid (gaps, insertions and deletions are equally penalized with -1), and for all valid matches, a new set of repeat and spacer is added to the array.

CRISPRDetect can extend a CRISPR beyond the sequence start and end positions
for circular genomes. It can handle cases where the sequence starts within a CRISPR array, splitting the array into two. If during prediction the genome start or end position is reached, instead of stopping the extension process immediately, CRISPRDetect continue analysing sequences from the end or start respectively. If an extension is favourable, CRISPRDetect uses the minimum number of bases required to complete the set of repeat and spacer, and stops the extension process (to avoid positional discrepancy). This was found multiple times in the current release of bacterial genomes (release date 11/14/2013).

![Figure 5.2 Extending CRISPRs](image)

**Figure 5.2 Extending CRISPRs.** A) Initial state of the CRISPR. B) CRISPR after extension operation. CRISPRs are extended on both ends by identifying degenerated repeats.

### 5.3.3.4 Remove degenerate repeats: shorten array

Degenerated repeats can be falsely predicted by extending the CRISPRs and may require manual intervention to correct them, specifically where the extension is true in one end and false in another. CRISPRDetect supports removal of degenerated repeats from either end, both automatically by using a minimum percentage similarity between the representative

---

repeat and individual repeats, as well as by specifying the number of repeats and side of the array. In automatic shortening, the operation stops when a repeat shows identity above the minimum percentage identity cutoff or the minimum number of repeats (default 2) specified for initial array prediction.

5.3.3.5 Trim repeat ends: shorten repeats

Initial prediction of repeats may consist of additional bases at the ends that belong to the spacers. This is due to the situations where the terminal base(s) of the spacers is near identical in an array. CRISPRDetect utilises a set of methods such as comparison with a known repeats library, known motifs (e.g. ATTGAAAN) found in the end of repeats, repeat end region degeneracy (default \(\geq 20\%\) base mismatch), to predict the true repeat/spacer boundary. In interactive mode, the users can trim one or both sides of the repeats by any number of bases, as long as the repeat retains the minimum word length specified in the parameters for initial array prediction.

5.3.3.6 Correct gap(s) at repeat ends

Due to extension of repeat length (i.e. extension on the 3’ and/or 5’ end of the repeats), followed by alignments and trimming; the repeats may end with one or more deleted or inserted bases. This makes the spacer longer and causes an incorrect spacer sequence. CRISPRDetect, identify the missing bases in the adjacent spacer (or flanking regions) and correct the gaps at the repeat end(s) for cases where the adjacent spacer length is equal or greater than the average spacer length. This is avoided where a portion of the repeat-spacer junction is deleted. If the gap(s) were caused due to insertion, the relative bases are moved to the insertion/deletion column improving both the array quality and spacer sequence(s).

5.3.3.7 Correct common insertion(s) in all the repeats of an array

During the alignment of the repeats gaps may be introduced, which can lead to a gap being
placed in the representative repeat (Fig. 5.3.A). However, gaps in the representative repeat are not biologically possible and can be misleading. Hence, the relative base(s) are moved to a dedicated column termed “Insertion/Deletion” (Fig. 5.3.B) along with the coordinate where the insertion occurred. The coordinates are fixed by comparing the non-gapped repeats and spacers against the input sequence.

**A.**

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
</tr>
</thead>
<tbody>
<tr>
<td>..................</td>
<td>~ ..................</td>
</tr>
<tr>
<td>GACGCAACCCTGACT</td>
<td>GTCCGGAATATCA</td>
</tr>
<tr>
<td>..................</td>
<td>~ ..................</td>
</tr>
<tr>
<td>GC ................</td>
<td>TGATCA</td>
</tr>
<tr>
<td>..................</td>
<td>~ ..................</td>
</tr>
<tr>
<td>ACCGATGGatat</td>
<td>AGATACGGCTTTAGGCTCTTCA</td>
</tr>
<tr>
<td>..................</td>
<td>~ ..................</td>
</tr>
<tr>
<td>ATTC ...........</td>
<td>TAAAACCGGGTTGGCAACCTGACG</td>
</tr>
<tr>
<td>..................</td>
<td>~ ..................</td>
</tr>
<tr>
<td>CGGTAT CCCCCGCTGGGG</td>
<td>ATCAC</td>
</tr>
</tbody>
</table>

**B.**

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
<th>Insertion/Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>..................</td>
<td>~ ..................</td>
<td></td>
</tr>
<tr>
<td>GACAGCAACCCTGACT</td>
<td>GTCCGGAATATCA</td>
<td>C, T [X1, X2]</td>
</tr>
<tr>
<td>..................</td>
<td>~ ..................</td>
<td></td>
</tr>
<tr>
<td>ACCGATGGatat</td>
<td>AGATACGGCTTTAGGCTCTTCA</td>
<td></td>
</tr>
<tr>
<td>..................</td>
<td>~ ..................</td>
<td></td>
</tr>
<tr>
<td>TAAAACCGGGTTGGCAACCT</td>
<td>G [X3], Deletion [X4]</td>
<td></td>
</tr>
<tr>
<td>..................</td>
<td>~ ..................</td>
<td></td>
</tr>
<tr>
<td>CGGTAT CCCCCGCTGGGG</td>
<td>ATCAC</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.3 Handling insertion(s) and deletion(s) in CRISPRDetect.** (A) The initial state. (B) Array after applying correction. The relative inserted bases are moved to Insertion/Deletion column. The position of the insertion/deletion is reported in the square brackets.

### 5.3.3.8 Identify mutated repeat(s) in sequences predicted to be long spacers

When a substantial portion of a repeat and/or a repeat-spacer junction got deleted, the repeats fail to retain the minimum percentage identity and end up being added to the next spacer, making the spacers unusually long. CRISPRDetect checks for such cases where the spacers are longer than median spacer length with a default (50%, can be specified by user to any positive integer >35) between the representative repeat and the whole spacer,
revealing not only partial repeat deletion, but also partial and/or total spacer loss.

5.3.4 Finding direction and predicting subtype

The direction of a CRISPR is calculated by the CRISPRDirection algorithm. The arrays with reverse direction are automatically reversed. Additionally when the representative repeat matches a known repeat with confirmed family information, the family of the repeat is recorded. If the representative repeat does not match any known repeats, and if annotation information is available, CRISPRDetect utilises the presence of Cas genes, to predict the family. This module is further utilized in correcting repeat boundaries, scoring and validation of the arrays.

5.3.5 Scoring the predictions

A scoring system gives each predicted array a score based on features of known biological and physical properties such as presence/lack of Cas genes in the genome, match to known experimentally proven repeats, presence of certain sequence motifs in the repeats, identity among the spacers, degeneracy among the repeats, as well as repeats and spacers length specific scoring methods are used. A final score for each of these CRISPRs are determined by summing all the scores from the individual methods. The complete scoring methods and scores can be found in Appendix C1. The CRISPRs with negative scores below -1.5 are discarded, and the remaining CRISPRs are further processed and listed in a position specific manner in the final output.

5.4 Results and Discussion

5.4.1 Overview:

CRISPRDetect was run with the default parameters on 2734 bacterial and archaeal
genomes from Genbank (5112 sequences). A typical genome of 4Mb can generate hundreds of putative arrays, and there were over 200,000 putative arrays (with word size 11 and minimum repetition of 2), which were then subject to refinement though the CRISPRDetect pipeline. A total of 4645 CRISPRs (4074 good arrays with score >=1.5) were found with score >0.25, minimum number of repeats >=3 and minimum repeat length>=21 in 1319 complete genomes. There were 3980 bacterial and 665 archaeal CRISPRs arrays.

We compared these to three previous programs, run with their default values. CRISPI was not used as it is online only and the algorithm is not well documented. CRT predicted 4097, PILER-CR 3802 and CRISPRFinder 4734 (Fig. 5.4). All four programs predicted 2706 arrays in common. These arrays are typically longer with multiple identical DRs. CRISPRFinder predicted the most arrays, in post processing many of these ~800 (as reported arrays in CRISPRdb is 3912) would have been tagged as questionable because they have one or more unusually long spacer(s). CRISPRDetect shows good concordance with PILER-CR and CRT (e.g. an additional 394 arrays in common between the three). In addition, CRISPRDetect predicted an additional 987 arrays (780 arrays with high confidence; score >=1.5). In addition to those shown in Fig. 5.4, CRISPRDetect predicted an additional 2088 arrays with score >0 and number of repeats =2.

---

48 The reported CRISPRs in CRISPRdb from a similar dataset are 3912 arrays (cf. 4734 predicted here), which are predicted by CRISPRFinder but also passed through post-processing. As those post-processing scripts were not available they could not be used here.
Figure 5.4 Comparison of the number of CRISPR predicted by four programs. Only arrays with minimum 3 repeats and a score >0.25 are compared.

The sizes of DRs are shown in Table 5.1. The distributions of the sizes are broadly similar to that previously shown or that in the current release of CRISPRdb (Grissa, et al., 2007). CRISPRdb includes repeats with a range limit of 23-55 nt and spacers 16-100 nt. These limits were applied heuristically at the inception of the database based on current knowledge, and avoidance of predicting 'questionable' arrays. We extend the concept of flagging questionable arrays, by providing a scoring scheme that gives an index of the likelihood that an array like genomic repeat can be a CRISPR.

The sizes of the repeats are mainly similar to that observed by Grissa et al 2007. Our analysis supports the idea that most DR fall into three size ranges (small 24-25, medium 28-30, and large 36-37) differently represented in archaea and bacteria. These all contain repeats similar to experimentally determined repeats\(^{49}\).

---

\(^{49}\) A list of 26 experimentally valid DRs can be found in (Biswas et al 2013)
Table 5.4 The length distribution of the representative repeats in CRISPRDetect output. 'Good' CRISPRs (i.e. score >= 1.5) are counted with only one strain per species. Peaks for each size range are in bold. The relative percentage for each DR length is shown in parenthesis.

<table>
<thead>
<tr>
<th>DR length</th>
<th>Bacteria</th>
<th>Archaea</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>0 (0)</td>
<td>1 (0.2)</td>
<td>1 (0.0)</td>
</tr>
<tr>
<td>50</td>
<td>1 (0.0)</td>
<td>0 (0)</td>
<td>1 (0.0)</td>
</tr>
<tr>
<td>49</td>
<td>2 (0.1)</td>
<td>0 (0)</td>
<td>2 (0.1)</td>
</tr>
<tr>
<td>48</td>
<td>3 (0.1)</td>
<td>0 (0)</td>
<td>3 (0.1)</td>
</tr>
<tr>
<td>47</td>
<td>8 (0.4)</td>
<td>0 (0)</td>
<td>8 (0.3)</td>
</tr>
<tr>
<td>46</td>
<td>7 (0.3)</td>
<td>0 (0)</td>
<td>7 (0.3)</td>
</tr>
<tr>
<td>45</td>
<td>2 (0.1)</td>
<td>0 (0)</td>
<td>2 (0.1)</td>
</tr>
<tr>
<td>41</td>
<td>4 (0.2)</td>
<td>0 (0)</td>
<td>4 (0.2)</td>
</tr>
<tr>
<td>40</td>
<td>7 (0.3)</td>
<td>1 (0.2)</td>
<td>8 (0.3)</td>
</tr>
<tr>
<td>39</td>
<td>12 (0.6)</td>
<td>1 (0.2)</td>
<td>13 (0.5)</td>
</tr>
<tr>
<td>38</td>
<td>29 (1.4)</td>
<td>0 (0)</td>
<td>29 (1.1)</td>
</tr>
<tr>
<td>37</td>
<td>278 (13.3)</td>
<td>50 (9.2)</td>
<td>328 (12.4)</td>
</tr>
<tr>
<td>36</td>
<td>311 (14.9)</td>
<td>13 (2.4)</td>
<td>324 (12.3)</td>
</tr>
<tr>
<td>35</td>
<td>53 (2.5)</td>
<td>3 (0.6)</td>
<td>56 (2.1)</td>
</tr>
<tr>
<td>34</td>
<td>12 (0.6)</td>
<td>0 (0)</td>
<td>12 (0.5)</td>
</tr>
<tr>
<td>33</td>
<td>34 (1.6)</td>
<td>2 (0.4)</td>
<td>36 (1.4)</td>
</tr>
<tr>
<td>32</td>
<td>143 (6.8)</td>
<td>4 (0.7)</td>
<td>147 (5.6)</td>
</tr>
<tr>
<td>31</td>
<td>53 (2.5)</td>
<td>30 (5.5)</td>
<td>83 (3.1)</td>
</tr>
<tr>
<td>30</td>
<td>457 (21.9)</td>
<td>167 (30.6)</td>
<td>624 (23.7)</td>
</tr>
<tr>
<td>29</td>
<td>347 (16.6)</td>
<td>36 (6.6)</td>
<td>383 (14.5)</td>
</tr>
<tr>
<td>28</td>
<td>233 (11.1)</td>
<td>16 (2.9)</td>
<td>249 (9.4)</td>
</tr>
<tr>
<td>27</td>
<td>22 (1.1)</td>
<td>4 (0.7)</td>
<td>26 (1.0)</td>
</tr>
<tr>
<td>26</td>
<td>19 (0.9)</td>
<td>17 (3.1)</td>
<td>36 (1.4)</td>
</tr>
<tr>
<td>25</td>
<td>15 (0.7)</td>
<td>107 (19.6)</td>
<td>122 (4.6)</td>
</tr>
<tr>
<td>24</td>
<td>22 (1.1)</td>
<td>84 (15.4)</td>
<td>106 (4.0)</td>
</tr>
<tr>
<td>23</td>
<td>6 (0.3)</td>
<td>7 (1.3)</td>
<td>13 (0.5)</td>
</tr>
<tr>
<td>22</td>
<td>6 (0.3)</td>
<td>1 (0.2)</td>
<td>7 (0.3)</td>
</tr>
<tr>
<td>21</td>
<td>4 (0.2)</td>
<td>1 (0.2)</td>
<td>5 (0.2)</td>
</tr>
<tr>
<td>Total:</td>
<td>2090</td>
<td>545</td>
<td>2635</td>
</tr>
</tbody>
</table>
Figure 5.5 CRISPRDetect predicted arrays from *Escherichia coli*. (A) CRISPR1. (B) CRISPR2. CRISPRDetect identified the direction of the transcription and represented the arrays correctly. Comparing the DR to known families of repeats identified the family.

Larger and smaller repeats are rare. However, a class of repeats of “extra large” 46-50 bases is well supported (first noted in *Bacteroides fragilis*; Grissa et al 2007), with
similar repeats found here in other bacterial genera (e.g. *Acetobacter*, *Fibrobacter* (48-49 nt), *Weeksella virosa* (50 nt)). All of these 47-50 nt long repeats have similar sequences particularly at the 3’ end. Although the CRISPRDetect algorithm predicted a few repeats >50 nt long, they have little support, and are tagged as questionable. There are 11 repeats from both bacteria and archaea with sizes <23.

The array with the most repeats (i.e. 592) is the marine bacterium *Haliangium ochraceum* with 36 base long repeats (and two arrays nearby of 190 and 37 with identical representative repeats). The process used by CRISPRDetect is illustrated by the analysing the well-studied *Escherichia coli* genome (e.g. DH10B, NC_010473, 4.6 Mb). It contained 52 possible arrays, on refinement two CRISPRs were identified at position around 2.9Mb. Initial directions of the arrays in the default output (without applying CRISPRDirection algorithm) were incorrectly oriented, and all three previous programs failed to accurately predict the repeat boundary for the second CRISPR2 (Fig 5.5A) missing a repeat base G on the 3’ end. The direction and the repeat boundaries were corrected by CRISPRDetect (Fig. 5.5B) by comparing the representative repeats of these CRISPRs to the known library of repeats, as well as identification of the likely CRISPR family (I-E, GAGTTCCCCGCGCCAGCGGGGATAAACCG) by the presence of signature Cas genes (Bhya, et al., 2011) in the genome.

5.4.2 Refinement of arrays internal structure

5.4.2.1 Insertion/deletion in repeats and spacers

Insertions and deletions can occur internally in repeats, these may be copied into new repeats during acquisition. CRISPRDetect has specific modules to address this issue, which is missing in all previous programs except PILER-CR. However, PILER-CR consistently fails to predict the cases where the repeat-spacer junction has one or more

---

50 DVD: Chapter5/Supplements/larger_arrays.txt
mutation/insertion within few bases (e.g. 6) at the end of repeat sequences (Supplement\textsuperscript{51} Case 1.A, 1.B, 1.C, 6.A, 6.B). The representation of insertion(s) can be simplified; as a single insertion in one repeat creates a gap insertion in all the other remaining repeats as well as the representative repeat. This also affects the relative coordinates of the following repeats and spacers (Supplement\textsuperscript{52} Case 1.D). Insertion of multiple bases towards the centre of a repeat can lead the program splitting the entire CRISPR into multiple short CRISPRs (Supplement\textsuperscript{53} Case 1.E), hence losing one or more spacers in the process.

All three programs support partial deletions in spacers. However, none of the programs supports identification and visualization of total spacer loss (Supplement\textsuperscript{54} Case 1.F). A better way to handle and represent the insertions and deletions is implemented in CRISPRDetect, where the insertion(s) in the repeats are shown together with their relative coordinate(s) as well as the deletions at the repeat-spacer junctions.

\subsection*{5.4.2.2 Identification of degenerated repeats in spacer sequence}

If repeats degenerate they may not be recognised in the initial analysis resulting in a long spacer. CRISPRDetect addresses this issue by searching the presence of partially deleted and/or mutated repeats with lower identity threshold (default 50\%, minimum 25\%) in all spacers with length greater than the median spacer length (Supplement\textsuperscript{55} Case 2.A, 4.A).

\textsuperscript{51} DVD: Chapter5/Supplements/case_studies.txt
\textsuperscript{52} DVD: Chapter5/Supplements/case_studies.txt
\textsuperscript{53} DVD: Chapter5/Supplements/case_studies.txt
\textsuperscript{54} DVD: Chapter5/Supplements/case_studies.txt
\textsuperscript{55} DVD: Chapter5/Supplements/case_studies.txt
Figure 5.6 Identification of a mutated repeat in spacer. (A) Initial CRISPR. (B) CRISPR after identification of degenerated repeat in the CRISPR.

5.4.2.3 Improving arrays by adjusting the repeat ends

CRISPRDetect introduces the use of a library of likely repeats (part of CRISPRbank) to automatically refine novel repeats. This can be used in both an automatic and interactive way. If the new representative repeat matches a known library repeat, then repeat sequence is adjusted to have the same length. In addition, if the representative repeat contains only a known repeat boundary motif (e.g. ATGAAA 3’), then the 3’ end of the repeat is adjusted (Fig. 5.5B). Additionally, the user has the option to interactively increase or decrease the repeat length in a predicted array.
Figure 5.7 Correction of repeat boundaries. (A) The representative repeat is found to match ATTGAAA(N) (Ref. Motif) and corrected the boundary. (B) The representative repeat aligned with a known repeat (Ref. repeat) and DRs are corrected using the reference repeat boundaries.

5.4.2.4 Identification of degenerated repeat and/or spacers beyond the end of an array

Repeats with large number of mutations may occur at the terminal repeats, making them hard to recognise. CRISPRDetect applies a lower threshold to extend arrays. The default settings are set quite stringently for the arrays that are held in the reference database (CRISPRBank\textsuperscript{56}). However this is user tuneable in both the automatic and interactive version of the program. This allows users to investigate the decay of CRISPRs and assists with orientating them (Supplement\textsuperscript{57} Case 3.A, 3.B).

Array extension is useful for considering closely spaced arrays, separated by deleted/degenerated repeats or insertions. CRISPRDetect supports an automatic or interactive array specific extension permitting detection with identity as low as 35%. It also

\textsuperscript{56} http://bioanalysis.otago.ac.nz/CRISPRBank  
\textsuperscript{57} DVD: Chapter5/Supplements/case_studies.txt
supports a ‘dynamic checking’ method, where instead of using the global representative repeat, the adjacent repeat is used as reference. The advantage of this method is that it allows dynamic adaptation where mutations have been propagated through part of the array (Supplement 58 Case 4.B).

5.4.2.5 False positive predictions from tandem repeats

Some types of tandem repeats may be mis-identified as CRISPR arrays. In this case the 'spacers' will be near identical. CRISPRFinder has dedicated function to avoid repeats that imposes a rule of dissimilarity within the spacers (Grissa, et al., 2007). However, some true arrays do have a number of exactly identical spacers, followed by few non-identical spacers (Supplement- Case 4.A, 5.A, 5.B). This issue has been addressed in CRISPRDetect by allowing an automatic extension of repeat length (both side) into spacers, but allowing this extension to extend over some non-identical columns (Supplement- Case 2.A).

5.4.2.6 Array Orientation

Previous tools did not predict orientation, resulting in ~50% with an incorrect orientation. To improve this issue, a recent development named CRISPRDirection with 94% accuracy has been adopted and applied. The arrays identified as wrongly orientated has been automatically adjusted (reverse complemented) accordingly. Out of the 4074 “good” arrays (Results 5.4.1), 2320 arrays found to have reverse orientation in the initial sequence.

5.4.2.7 Incorporation of comparisons to an internal database

As yet there are no dynamically interactive CRISPR prediction tools, and the given parameters are globally applied to all the CRISPRs in the input sequence. Parameters cannot be changed once the arrays are predicted without completely re-running the prediction program. Although, CRISPRFinder is supported by post-processing tools and

---

58 DVD: Chapter5/Supplements/case_studies.txt
database (i.e. CRISPRdb) as is CRISPI, the CRISPR predictions are not complemented by prior knowledge or an intercommunication between the prediction program and the database is not available. Such methods could greatly benefit the prediction of true positive CRISPRs especially the shorter or degenerated ones by querying the representative repeat or spacers against a database of true CRISPRs. CRISPRDetect address this issue (in the CRISPRDetect web-server), by incorporating a database of precomputed valid CRISPRs (CRISPRBank) from all published bacterial and archaeal genomes.

Users can validate all newly predicted CRISPRs with a minimum likelihood scores (default 1.5) during initial prediction or once the output is generated. The representative repeat or each spacer can be directly searched in the database showing occurrences in other published genomes without leaving the prediction program. CRISPRBank contains over 6000 CRISPRs (arrays with minimum 2 repeats and score >0.25) with detail information including family, direction and likelihood scores. Apart from its use in CRISPRDetect prediction, we also made CRISPRBank available online with different searching mechanism. For example, it is possible to retrieve all related arrays that share a common family, repeat, spacer or other properties like number of repeats, likelihood score etc.

5.4.2.8 CRISPR family information

In CRISPRDetect the type/family of a CRISPR is identified using two techniques. Firstly by using sequence similarity against known repeat families and secondly by utilizing the presence of specific Cas genes in the annotated genome (files with extension .gbk). First priority is given to the repeat match, and if the type/family couldn’t be predicted from the first method, the type/family is determined based on the presence of the Cas genes in the genome. CRISPRDetect, also lists the Cas genes present in the genome, together with the set of signature Cas genes that were used to determine the type of the CRISPR. If multiple instances of a Cas gene are found, then the closest one is reported.
5.4.2.9 Identification of true positive short arrays

It might be possible for minimal CRISPR arrays with only 2 repeats and one spacer to function (Supplement file short_arrays.txt). CRISPRFinder flags such CRISPRs as questionable arrays. CRISPRDetect addresses this issue by comparing the representative repeat to a known repeat library, followed by attempting to find degenerate repeats at either end to extend the array. CRISPRDetect predicted over 750 such short arrays with score > 1.5. About 70% of all these arrays were not predicted by CRT or PILER-CR (with default parameters). As they may introduce many false positives, PILER-CR no longer supports prediction of CRISPRs with 2 repeats.

5.4.2.10 Scoring arrays

The quality of the final prediction is scored by a set of rules. CRISPRDetect scores each array with nine different CRISPR properties that includes both positive (e.g. length of repeat) and negative scores (e.g. absence of Cas genes in the genome) (Appendix C1). Arrays with scores below the default 1.5 (or user given cutoff score) are flagged as questionable. In the final output, arrays with scores below 0 are not reported. These parameters and adjustable in both the automatic and interactive version.

5.4.3 Supplementary analysis

Spacers, from the CRISPRDetect output (webpage) can be sent directly to CRISPRTarget to facilitate the prediction of targets of spacers in foreign DNA (e.g. the bacteriophage division of GenBank) (Biswas, et al., 2013). CRISPRTarget uses a flexible algorithm that takes the formatted and predicted spacer sequences from CRISPRDetect (also accept other formats) and uses these to search databases for likely targets.

CRISPRDetect shows any repeats that have an exact match in CRISPRBank. If desired, these DR can be further analysed by CRISPRmap, which can classify the DR repeats based on sequence and structural similarity into one of 40 families or 33 structured
motifs (Lange, et al., 2013). This can then be used to predict the phylogenetic distribution of the family that the DR matches.

5.5 Conclusion

CRISPRDetect is currently the most advanced CRISPR prediction tool, which was designed to address known limitations, but also provides additional information missing in the previous generation of CRISPR prediction tools. Appendix C.2 shows a comparison of features and improvements over the existing tools.

This study is focused on the prediction of CRISPRs by analysing both the CRISPR properties and distinguishing these from 'CRISPR like’ repeats which can easily be predicted incorrectly as a CRISPR. CRISPRDetect, in combination with CRISPRBank and CRISPRTarget, now provides a comprehensive resource for the detection and analysis of CRISPRs. The enhanced annotation of arrays to reveal orientation, precise repeat-spacer boundaries, small and large mutations (substitution, deletion and insertions) in spacers and repeats and other features, can be interrogated using a web interface, or be incorporated into genome annotation pipelines for improved annotation.

Recently there has been the extension of CRISPR prediction to metagenomic data. The results of these analysis are not single CRISPR arrays, but complex models of cross organism CRISPRs. One method extended CRT to metagenomic data (metaCRT) (Rho, et al., 2012), the other one is a novel algorithm (CRASS) which in its initial steps is similar to the algorithm used here (detecting two identical 8 base words within windows of varying lengths) (Skennerton, et al., 2013). However biological questions addressed by metagenomic analysis is quite different to the CRISPRDetect presented here. A modified version of CRISPRDetect (with smaller word length and identity cutoff) can be applied to metagenomic data after a de novo assembly.
Chapter 6

Conclusion

The work presented in this thesis has demonstrated how in-silico approaches can be efficient and effective at identifying functional regulatory elements in eukaryotes that play important roles in post-transcriptional gene regulation. A typical search for different types of regulatory elements in any 3’ UTR produces long lists of sequence motifs and binding sites. However, not all of them are functional or work in the same way. This raises an important question of how to determine if any two of these elements are functionally related or not. Scan for motifs can be useful in determining how close elements need to be to affect each other’s functions. Often, the elements may need to be in the immediate neighbourhood of one another, where for others, it may play its role from a distant location. However, not all the regulatory elements found in the same UTR may directly affect the closest binding site. This study provides a tool that can generate hypotheses that will lead to further testing.

Extensive analysis has been done to develop a number of novel tools and techniques to enable CRISPR specific studies in ways that were not available before. CRISPRTarget provides researchers the most advanced tool for finding the true targets of any CRISPR RNA. The most widely practiced method for finding the target(s) of a CRISPR was previously simple BLAST searching, where the spacers were searched against NCBI non-redundant nucleotide database. However, as the typical length of the spacers is short (<35 nt), a general BLAST screen can easily result in thousands of hits. Finding the true targets from this list was difficult and time consuming. CRISPRTarget is also the first tool of this kind that takes the output from all three major CRISPR array prediction tools. In addition to supporting CRISPR arrays as input, it also accepts a list of spacers in multiFASTA format. The existing prediction tools were developed on average ~6 years ago and do not include
recent refinements or direction of CRISPR transcription, which can be critical for finding true repeat and spacer boundaries. CRISPRTarget efficiently addressed these issues by providing sophisticated scoring techniques that is powered with CRISPR type-specific properties. Due to its simple, robust and efficient identification capability, CRISPRTarget already attracted researchers across the world and been cited a number of times since the publication in early 2013.

As stated earlier, the existing prediction tools do not include prediction of direction of CRISPR transcription, which can be critical in various ways including finding the true targets of a CRISPR/Cas system. To address this issue the CRISPRDirection algorithm was developed, which is again the first method of this kind that can be added to any CRISPR prediction and annotation pipeline. CRISPRDirection demonstrates how various easily calculable parameters can work together to predict the direction with high sensitivity, without requiring prior knowledge of the species type or family information of the CRISPRs. However, the CRISPRDirection method can be easily be modified to analyse specific genomes, for example incorporating genome specific biological knowledge such as AT% or repeat sequence motifs.

During the extensive analysis involved with CRISPRTarget and CRISPRDirection, the limitations of the other existing tools were explored. This led to the development of CRISPRDetect, as yet the most advanced CRISPR prediction and validation tool. For a long time CRISPRs were predicted by genomic repeat/motif finding tools. Many existing CRISPR specific prediction methods still rely mainly upon just sequence identity in repeats and dissimilarity across the spacers. These rules can be effective in predicting longer CRISPRs with few or no mismatches or deletions or insertions, but shorter arrays with more degeneracy needed to be treated carefully. Additionally, biological information should be incorporated to validate the CRISPRs. CRISPRDetect made use of the knowledge provided by CRISPRDirection to improve detection of the repeat and spacer boundaries, family, orientation and added ~15 different improvements over the existing prediction tools. Its accurate array specific analysis will give researchers an easy way to validate any CRISPR. The added capabilities like direct query against CRISPRBank showing the
presence of repeats and spacers in other organisms, more detailed output including presence
and position of Cas genes, and output in GFF format will make it readily useful for both
individual prediction and annotation pipelines. CRISPRDetect's highly interactive interface
supplemented with a wide range of refinement modules and intercommunication capability
to other CRISPR specific tools and resources will set the standard for future CRISPR
detection methods.

Future Directions

The bioinformatics tools described in this thesis cover two important areas of research
namely; translational control through various types of REs in eukaryotes, and CRISPR/Cas
systems in prokaryotes.

An interesting application of Scan for motifs will be to investigate the effective
distances (minimum, maximum) between different types of REs such as miRNA/protein
binding sites and other sequence motifs in human 3’ UTRs. By analysing certain 3’ UTRs
with true positive (experimentally verified) elements, element specific reference libraries
(distance properties) can be made. This is a multi-step process, involving; scanning entire
3’ UTRs cataloguing the positions of each element, followed by type specific comparisons.
This analysis may reveal many novel sequence motifs associated with translational control.

The CRISPR/Cas system specific tools can be applied to various novel studies. An
extension of the CRISPRTarget project would be to construct a database of potential
CRISPR targets (protospacers). This can be done by extracting the valid spacers from all
bacterial and archaeal CRISPRs, predicted by CRISPRDetect (Chapter 5), and searched
across various phage sequence databases.

The novel algorithm described in Chapter 4 for accurate prediction of the
transcribed strand of CRISPR, can be utilised to identify potential true leader regions. This
leader sequences can be analysed to identify different promoter elements that are specific to
different type/family of CRISPRs.

A modified version of CRISPRDetect can be applied to various environmental metagenomic sequence databases to identify potential new CRISPRs. These CRISPRs can be stored in a new section of CRISPRBank database for public access. Additionally, an enhancement to the CRISPRDetect will be the incorporation of potential CRISPR target information.


Science, 321, 960-964.


in subtype I-C/Dvulg CRISPR-Cas system, *Structure*, 20, 1574-1584.


Rinn, J.L., et al. (2007) Functional demarcation of active and silent chromatin domains in


Tili, E., et al. (2007) Modulation of miR-155 and miR-125b levels following lipopolysaccharide/TNF-alpha stimulation and their possible roles in regulating the


Appendix A

Supporting tables and contents for Chapter 2
A.1. Table UTR_Sequences

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
<th>Privileges</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>mTR_Family</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene_ID</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene_Symbol</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Species_ID</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTR_start</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTR_end</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSA_start</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSA_end</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seed_match</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCT</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.1 MySQL schema of the table UTR_Sequences.

A.2. Table Predicted_Targets_Info

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
<th>Privileges</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>mTR_Family</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene_ID</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene_Symbol</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Species_ID</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTR_start</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTR_end</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSA_start</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSA_end</td>
<td>int(15)</td>
<td>(NULL)</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seed_match</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCT</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>YES</td>
<td>(NULL)</td>
<td>(NULL)</td>
<td>select, insert, update, references</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.2 MySQL schema of the table Predicted_Targets_Info.
Appendix B

Supporting tables and contents for Chapter 4
B.1. Reference set of repeats

The list of 26 experimentally verified repeats, references given in the brackets

>GATAATCTCTTATAGAATTGAAAG
>Cthe I-B (Biswas et al 2013)
>GTITTTTATGTACCCTATGAGGAATTTGAAAC
>Hvol_a I-B (Biswas et al 2013) GTTTCAGAGAACCTTGTGGGATTTGAAAC
>GTTCAGGGAACCTTGTGGGATTTGAAAC
>Hvol-b I-B (Biswas et al 2013) GTTTCAGGGAACCTTGTGGGATTTGAAAC
>GTTCAGGGAACCTTGTGGGATTTGAAAC
>Hvol_c I-B (Biswas et al 2013) GTTTCAGGGAACCTTGTGGGATTTGAAAC
>Ssol I-B (Biswas et al 2013)
>GATTAATCCCAAAGGAAATTGAAAG
>Xory I-C (Biswas et al 2013)
>GTGCGGTCTTACCGGGGCGTGTTGAATTTGAAAC
>Ecol_a I-E (Biswas et al 2013) GWGTCCCCGGCCAGCGGGGATAAAACCG
>GAGTCCCCGGCCAGCGGGGATAAAACCG
>Ecol_b I-E (Biswas et al 2013) GWGTCCCCGGCCAGCGGGGATAAAACCG
>GTGTTCCCCGGCCAGCGGGGATAAAACCG
>Patr I-F (Biswas et al 2013)
>GTTCAGGGAACCTTGGGGATTTGAAAC
>Sthe I-F (Biswas et al 2013)
>GTTCAGGGAACCTTGGGGATTTGAAAC
>Mmar I-B (Richter, et al., 2013)
>CTAAAAAGAATAACTTGGAAAATAAACAAGCATTTGAAAC
>SynPCC_CRISPR1 I-D (Scholz, et al., 2013)
>CTTTCTCTACTAATCTCGCGATCGGGACTGAAAC
>Spyo II-B (Biswas et al 2013)
>GTITTTAGAGCTATGCTTTGAAATGGGTTTTGAAAC
>Nlac (Zhang, et al., 2013) II-C
>GTGTTAGCTTCCCTTCTATTCAATGCTCAAAC
>Cjej II-C (Zhang, et al., 2013, Dugar, et al., 2013)
>GTITTTAGCCCTTTTTTAAATTTCTTTATGGTAAAAT
>Pfur III-B (Biswas et al 2013)
>GTTCCTAATAAGACTAAATAGAATTGAAAG
>Sepi III-A (Biswas et al 2013)
>GATCGATACCCACCCGAAGAAAGGAGGCAGGAAAC
>SynPCC_CRISPR2 III NA (Scholz, et al., 2013)
>GTTCACACCCCTTTTTCCCCTCGGGACTGAAAC
>Hbut NA-NA(Kenchappa, et al., 2013)
>GAACACTGAAAAGAGAATTGCAAG
>Mjan NA-NA(Beloglazova, et al., 2011)
>ATAAAATGAGACCTGTTTGGAATTGAAAT
>CDif I-B (Soutourina, et al., 2013)
>GTITTTATTAATCTAACTTGAATGTTGAATGAAAG
>Tien_TTX_6 I-A (Plagens, et al., 2012)
>GAACTCTTAAAGAGGGGATTGAAAG
>Tien_TTX_2 I-A (Plagens, et al., 2012)
>GTGAAAACTGAAAAGATGTAATGAAAC
>Cthe A I-B (Biswas et al 2013)
>GTGGAAGATTTGACTCCTCCAGTAAAACAGGATTGAAAC
>SynPCC_CRISPR3 III (Scholz et al 2013)
>GTTCCTAATCGGAGAAATTTGAGATTGAAAC
B.2. Flow diagram of processes in CRISPRDirection.pl

Figure B.1 Flow diagram of CRISPRDirection algorithm. No predictions are referred by NP.
B.3. Repeat classes covered in the list of reference repeats

Figure B.2 CRISPRMap output for 120 unique repeat sequences. The 120 reference sequences (Supplement59 Table 8) were searched against the sequences in the CRISPRMap database. Matches to diverse classes are shown as red lines.

59 DVD: Chapter4/Supplements/file1.docx
B.4. Combinations of analysis (leaving out each in turn) for 100 random arrays

Figure B.3 Prediction leaving out specific methods. Each individual method was left out and the combined PPV calculated. The greatest effect is seen when degeneracy is left out (from 0.95 to 0.87). Leaving out combinations (e.g. all those related to flanks) had greater effects (to 0.83).
### Table B.1 Combinations of analysis (leaving out each in turn) for 100 random arrays.

<table>
<thead>
<tr>
<th>Analysis skipped</th>
<th>SAMPLE</th>
<th>TP</th>
<th>TP (SD)</th>
<th>TP (SE)</th>
<th>FP</th>
<th>FP (SD)</th>
<th>FP (SE)</th>
<th>NP</th>
<th>NP (SD)</th>
<th>NP (SE)</th>
<th>PPV</th>
<th>PPV (SD)</th>
<th>PPV (SE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motif (ATTGAAA) search</td>
<td>100</td>
<td>87</td>
<td>2.9</td>
<td>0.31</td>
<td>10</td>
<td>2.73</td>
<td>0.86</td>
<td>3</td>
<td>1.3</td>
<td>0.75</td>
<td>0.90</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>A&amp;T ratio in repeats</td>
<td>100</td>
<td>93</td>
<td>2.02</td>
<td>0.21</td>
<td>6</td>
<td>1.86</td>
<td>0.76</td>
<td>1</td>
<td>0.77</td>
<td>0.77</td>
<td>0.94</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>AT Dist. In flanks</td>
<td>100</td>
<td>92</td>
<td>2.59</td>
<td>0.27</td>
<td>6</td>
<td>2.11</td>
<td>0.86</td>
<td>2</td>
<td>1.3</td>
<td>0.92</td>
<td>0.94</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Degeneracy in array</td>
<td>100</td>
<td>88</td>
<td>2.42</td>
<td>0.26</td>
<td>9</td>
<td>2.19</td>
<td>0.73</td>
<td>4</td>
<td>1.53</td>
<td>0.77</td>
<td>0.91</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Secondary Structure</td>
<td>100</td>
<td>89</td>
<td>2.96</td>
<td>0.31</td>
<td>9</td>
<td>2.7</td>
<td>0.9</td>
<td>2</td>
<td>1.14</td>
<td>0.81</td>
<td>0.91</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Longer leader search</td>
<td>100</td>
<td>92</td>
<td>2.1</td>
<td>0.22</td>
<td>6</td>
<td>1.8</td>
<td>0.73</td>
<td>2</td>
<td>1.25</td>
<td>0.88</td>
<td>0.94</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Motif, A-T ratio, Secondary Str.</td>
<td>100</td>
<td>83</td>
<td>3.5</td>
<td>0.38</td>
<td>17</td>
<td>3.5</td>
<td>0.85</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.83</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>AT Dist., Degeneracy &amp; longer leader</td>
<td>100</td>
<td>81</td>
<td>3.9</td>
<td>0.43</td>
<td>12</td>
<td>3.05</td>
<td>0.88</td>
<td>7</td>
<td>2.38</td>
<td>0.9</td>
<td>0.87</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td><strong>None</strong></td>
<td><strong>100</strong></td>
<td><strong>94</strong></td>
<td><strong>2.14</strong></td>
<td><strong>0.22</strong></td>
<td><strong>5</strong></td>
<td><strong>2.06</strong></td>
<td><strong>0.92</strong></td>
<td><strong>1</strong></td>
<td><strong>0.82</strong></td>
<td><strong>0.82</strong></td>
<td><strong>0.95</strong></td>
<td><strong>0.02</strong></td>
<td><strong>0.02</strong></td>
</tr>
</tbody>
</table>

SD: Standard Deviation;
SE: Standard Error;
B.5. Relative AT richness in the two flanks

Figure B.4 Relative AT richness in the two flanks. AT% for windows of the specified lengths were calculated on either flank, the side with the higher AT% was used as a predictor of the leader. Predictions for different AT% are shown (1-14%). TP: True positives, FP: False positives, NP: No Prediction, PPV: Positive Predictive Value. Data are the mean and standard deviation of 100 random samples of 100 from the reference set of 460 arrays. The 10% difference and a window of 45 is used for normal specificity and 14% difference, with a window of 120 for high specificity.
Appendix C

Supporting tables and contents for Chapter 5
C.1. **Array scoring scheme**

1. Absence of Cas1 and Cas2 gene in the genome is penalised ( -1).
2. Match to known repeat with a small number of mismatches, using CRISPRDirection reference repeats with up to 7 mismatches (+3).
3. Repeat has at least 23 bases and ATTGAAA(N) at the end (+3)
4. Overall array repeats identity calculated by the following method (max. value +1)
   \[
   \text{(Average } \% \text{ identity of the repeats in the array - 80)/20}
   \]
5. If two or more repeats are not identical, ( -1.5)
6. Score for representative repeat length is calculated from the distribution of repeat length (Table C.1; Kunin et al., 2007) for all representative repeat within specified range. All repeats, within the range (i.e. 23-50) gets minimum +0.20 plus a length specific score calculated from the following table by repeat_length/highest peak (shown in bold).

Repeats outside range are penalized with:

\[-0.25\times(\text{Closest DR boundary} - \text{repeat_length})\quad \text{[when DR length <23]}\]

or

\[-0.25\times(\text{repeat_length} - \text{Closest DR boundary})\quad \text{[when DR length >50]}\]

Maximum negative score limit is -3, and positive score limit is +1.

7. Score for individual spacer length that belongs to 28-48 (Table C.2) are obtained using the following formula:

\[\text{spacer}_\text{score}=\frac{\text{spacer}_\text{count for the current spacer length}}{\text{highest}_\text{count}}\]

Any spacer with length outside this range is penalized by the following rule:

\[-0.10\times(\text{closest spacer boundary} - \text{spacer_length})\quad \text{[when spacer length <28]}\]

or

\[-0.10\times(\text{spacer_length} - \text{closest spacer boundary})\quad \text{[when spacer length >48]}\]

Finally, an average spacer score for the CRISPR is calculated using

\[\text{Average score}=\frac{\text{Sum\_of\_scores}}{\text{no\_of\_spacers}}\]
The maximum negative score limit is -3.

1. Overall spacer identity (check if all spacers belong to a single cluster with maximum allowed identity 80%): Negative scoring (if all spacers forms one cluster) with -3; positive score (0.25 * No_of_cluster); Maximum score limit +1.

2. Score total repeats using the formula:

   \[ \log(\text{no_of_perfect_repeats}) - \log(\text{no_of_degenerated_repeats}) \]

   The maximum negative score limit is -3.
**Table C.1** DR length distribution from (Kunin et al., 2007). The field with * are estimated from CRISPRdb records.

<table>
<thead>
<tr>
<th>DR length</th>
<th>Peak (Bacteria)</th>
<th>Peak (Archaea)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>0</td>
<td>15 (*)</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>25</td>
<td>6</td>
<td>37</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>28</td>
<td>67</td>
<td>7</td>
</tr>
<tr>
<td>29</td>
<td>68</td>
<td>9</td>
</tr>
<tr>
<td>30</td>
<td>51</td>
<td>46</td>
</tr>
<tr>
<td>31</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>32</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>36</td>
<td>69</td>
<td>9</td>
</tr>
<tr>
<td>37</td>
<td>55</td>
<td>14</td>
</tr>
<tr>
<td>38</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>49</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table C.2** Spacer length distribution.

<table>
<thead>
<tr>
<th>Spacer length</th>
<th>Spacer count</th>
<th>(Each '<em>' represents 179 occurrences,'+' represents 1/2 '</em>')</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>38</td>
<td></td>
</tr>
</tbody>
</table>

*Continued to the next page*
<table>
<thead>
<tr>
<th>Spacer length</th>
<th>Spacer count</th>
<th>(Each '<em>' represents 179 occurrences,'+' represents 1/2 '</em>')</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>199</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>4673</td>
<td>************</td>
</tr>
<tr>
<td>35</td>
<td>7388</td>
<td>******************</td>
</tr>
<tr>
<td>36</td>
<td>10717</td>
<td>##################################################################</td>
</tr>
<tr>
<td>37</td>
<td>8214</td>
<td>##################################################################</td>
</tr>
<tr>
<td>38</td>
<td>4848</td>
<td>##########################################################################</td>
</tr>
<tr>
<td>39</td>
<td>3079</td>
<td>**********</td>
</tr>
<tr>
<td>40</td>
<td>2260</td>
<td>**********</td>
</tr>
<tr>
<td>41</td>
<td>1496</td>
<td>**********</td>
</tr>
<tr>
<td>42</td>
<td>1076</td>
<td>**********</td>
</tr>
<tr>
<td>43</td>
<td>821</td>
<td>*<em>.</em></td>
</tr>
<tr>
<td>44</td>
<td>601</td>
<td>***</td>
</tr>
<tr>
<td>45</td>
<td>387</td>
<td>**</td>
</tr>
<tr>
<td>46</td>
<td>273</td>
<td>{*}</td>
</tr>
<tr>
<td>47</td>
<td>163</td>
<td>'+'</td>
</tr>
<tr>
<td>48</td>
<td>101</td>
<td>'+'</td>
</tr>
<tr>
<td>49</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

*Continued to the next page*
Table C2: continued from the previous page

<table>
<thead>
<tr>
<th>Spacer length</th>
<th>Spacer count</th>
<th>(Each ‘<em>’ represents 179 occurrences, ‘+’ represents 1/2 ‘</em>’)</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

C.2. Comparison of CRISPREDetect with existing CRISPR prediction tools.

Table C3 shows a comprehensive comparison of CRISPRTarget with existing tools.
Table C.3 Comparison of four widely used CRISPR prediction tools with CRISPRDetect

<table>
<thead>
<tr>
<th>Feature</th>
<th>PILER-CR</th>
<th>CRT</th>
<th>CRISPI</th>
<th>CRISPRFinder</th>
<th>CRISPRDetect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allows insertion/deletions in specific repeats and spacer</td>
<td>yes</td>
<td>no</td>
<td>n/a</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Identifies complete spacer deletions</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Determines the direction</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Predicts type</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Identifies degenerate repeats in putative spacer sequence</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Identification of degenerated repeat and/or spacer in flanking regions</td>
<td>yes</td>
<td>yes</td>
<td>n/a</td>
<td>yes (with dedicated function)</td>
<td>yes (with dedicated function)</td>
</tr>
<tr>
<td>Annotates arrays at the end of circular genomes</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Filters tandem spacerless repeats</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
<td>yes (with limitation)</td>
<td>yes</td>
</tr>
<tr>
<td>Extends arrays with a lower stringency. Joins closely spaced arrays separated by degenerated repeats</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Trims poor repeats from arrays</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Compares to a database of known repeats and features</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Identification of arrays with only 2 repeats</td>
<td>no</td>
<td>yes</td>
<td>n/a</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Shows flanking regions</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Determines the representative repeat</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Shows Cas genes present in the genome</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Interactive web interface</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Pipeline version</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes/no</td>
<td>yes</td>
</tr>
<tr>
<td>Supporting database</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
C.3. Examples of the effects of various modules.

A.

<table>
<thead>
<tr>
<th>Flank</th>
<th>Repeat</th>
<th>Spacer</th>
</tr>
</thead>
<tbody>
<tr>
<td>TATGC</td>
<td>GGA A</td>
<td>GGACCGCAACCCGGTGCGRATACGACT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCACACGGCATCCGACGCTGAGGAG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GATGATCCGCRGCTGATTTTAGCTAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AGATAACGCTTTACGCTGCTCTAATTAAAGGTA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TAAAACGCCCTTCGCAACGC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCTAAACCTCAA</td>
</tr>
<tr>
<td>CGTTTAATCCCGCTGCGGCGGACAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B.

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
<th>Insertion/Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AA [X1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A [X1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A [X2]</td>
</tr>
<tr>
<td>CGTTTAATCCCGCTGCGGCGGACAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure C.1 Fixing gaps at the repeat ends

A.

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
<th>Rep. repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTCAGGCAACCCGGGACTTCAAT</td>
<td>GGACGCAACCCGGTGCGRATACGACT</td>
<td></td>
</tr>
<tr>
<td>TCACACGGCATCCGACGCTGAGGAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GATGATCCGCRGCTGATTTTAGCTAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGATAACGCTTTACGCTGCTCTAATTAAAGGTA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAAAACGCCCTTCGCAACGC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCTAAACCTCAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGTTTAATCCCGCTGCGGCGGACAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B.

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
<th>Rep. repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>GGGACGCAACCCGGTGCGRATACGACT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCACACGGCATCCGACGCTGAGGAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GATGATCCGCRGCTGATTTTAGCTAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGATAACGCTTTACGCTGCTCTAATTAAAGGTA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAAAACGCCCTTCGCAACGC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCTAAACCTCAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGTTTAATCCCGCTGCGGCGGACAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure C.2. Improved repeat and spacers by correcting the repeat boundaries
Figure C.3 Removal of degenerated repeats at either end of CRISPRs

A.

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A...C...T...</td>
<td>ATATGCAA</td>
</tr>
<tr>
<td>.............</td>
<td>GGCAGCAACCCTGGTTGGAT</td>
</tr>
<tr>
<td>.............</td>
<td>TCACCCGCAAATCCGCAATCG</td>
</tr>
<tr>
<td>.............</td>
<td>GAGTAATGGCAGGGGGGTAATT</td>
</tr>
<tr>
<td>.............</td>
<td>AGATACGCCCTTATGCACTGCT</td>
</tr>
<tr>
<td>.............</td>
<td>CAAAACACCGMGTCGCAACC</td>
</tr>
<tr>
<td>.............</td>
<td>TCTAAATTCAGATAGGAAAG</td>
</tr>
<tr>
<td>A...C.........</td>
<td>AAATGCATGAGGAGGAGG</td>
</tr>
</tbody>
</table>

Figure C.4 Handling insertion and deletions in arrays

A.

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
<th>Insertion/Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>.............</td>
<td>ATGACAGACCGAGCTGGTCGCCGAT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>CACCCCAACCCTGGTTGGAT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>ACCGGAAATCCGAAATCCGCGCCGGGAC</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>GAGTAATGGCAGGGGGGTAATT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>AGATACGCCCTTATGCACTGCT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>CAAAACACCGMGTCGCAACC</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>TCTAAATTCAGATAGGAAAG</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>AAATGCATGAGGAGGAGG</td>
<td>Deletion [X]</td>
</tr>
</tbody>
</table>

B.

<table>
<thead>
<tr>
<th>Repeat</th>
<th>Spacer</th>
<th>Insertion/Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>.............</td>
<td>ATGACAGACCGAGCTGGTCGCCGAT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>CACCCCAACCCTGGTTGGAT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>ACCGGAAATCCGAAATCCGCGCCGGGAC</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>GAGTAATGGCAGGGGGGTAATT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>AGATACGCCCTTATGCACTGCT</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>CAAAACACCGMGTCGCAACC</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>TCTAAATTCAGATAGGAAAG</td>
<td></td>
</tr>
<tr>
<td>.............</td>
<td>AAATGCATGAGGAGGAGG</td>
<td>Deletion [X]</td>
</tr>
</tbody>
</table>

C | T [X1, X2] | G [X1], Deletion [X] |
Appendix D

DVD table of contents
D.1. Electronic copy of the thesis

Ambarish_Biswas_PhD_thesis_Dec2013.pdf

D.2. The perl programs for Chapter 2

- Chapter2/Scripts/sfm_main.pl

D.3. The MySql table for Chapter 2

- Chapter2/Tables/gene_info.sql

D.4. The perl programs for Chapter 3

- Chapter3/CRISPRTarget/crispr_target.pl

D.5. Other executables for Chapter 3

- Chapter3/CRISPRTarget/Scripts/pilercr
- Chapter3/CRISPRTarget/Scripts/CRT1.2-CLI.jar

D.6. Supplementary files for Chapter 3

- Chapter3/Supplements/file1.txt
- Chapter3/Supplements/file2.html
- Chapter3/Supplements/file3.txt
- Chapter3/Supplements/file4.xlsx
D.7. The perl programs for Chapter 4

- Chapter4/Scripts/CRISPRDirection_v1.0.zip

D.8. Supplementary files for Chapter 4

- Chapter4/Supplements/file1.docx
- Chapter4/Supplements/all_3571_arrays.txt
- Chapter4/Supplements/file2.xls

D.9. The perl programs for Chapter 5

- Chapter5/Scripts/CRISPRDetect_v1.0.pl

D.10. Supplementary files for Chapter 5

- Chapter5/Supplements/case_studies.txt
- Chapter5/Supplements/larger_arrays.txt
- Chapter5/Supplements/short_arrays.txt
Glossary

**Basic Local Alignment Search Tool (BLAST).** The BLAST algorithm, developed by Stephen Altschul, Warren Gish and David Lipman at the National Center for Biotechnology Information (NCBI). It compares DNA or protein sequences for sequence alignments and is a faster (but less sensitive) alternative to Smith-Waterman algorithm.

**Cas genes.** The genes that code for proteins related to the CRISPRs are known as Cas (CRISPR-associated) genes.

**CASCADE (CRISPR-associated complex for antiviral defense).** CASCADE is a 405-kDa ribonucleoprotein complex comprising multiple Cas proteins (CasA1B2C6D1E1) and typically a 61 nucleotide CRISPR RNA (crRNA) with 5′-hydroxyl and 2′,3′-cyclic phosphate termini.

**Cis-Regulatory Elements (CREs).** CREs are regions of non-coding DNA that regulate the transcription of the nearby genes. The Latin prefix *cis* means ‘on the side’ and the CREs are found in the vicinity of the gene(s) they regulate.

**Database.** A database is an organized collection of data stored in specific structures.

**Expect (e) Value.** The term e-value refers to the value of a random variable one would expect to find in a set of weighted average of all possible values.

**False Discovery Rate (FDR).** It is a statistical method used for testing multiple hypothesis. For a dataset where predictions are either true or false, the FDR is the percentage of false predictions in the dataset.

**Motif.** A motif is DNA, RNA or protein sequence pattern that believed to have important biological significance.
**Positive Predictive Value.** It is a statistical method used for reporting the amount of true positive predictions in a set of both true positive and true negative predictions. For a data set containing both true positive predictions and true negative (false) predictions, the FDR score can be obtained by \( \frac{TP}{TP+FP} \), where TP= True positive prediction, and FP= True negative or false prediction. PPV ranges between 0 to 1.

**Position Weight Matrix (PWM).** The PWM also known as position-specific scoring matrix (PSSM) is a commonly used numerical representation of motifs in biological sequences.

**Reference Sequence (RefSeq) database.** The NCBI RefSeq database contains a comprehensive set of well-annotated, non-redundant sequences, including genomic DNA, transcripts and proteins.

**RNA interference (RNAi).** The regulatory pathways mediated by small RNAs such as miRNA and siRNAs (Short Interfering RNAs) to inhibit gene expression are collectively referred as RNAi or RNA silencing.

**Temperate phage.** Bacteriophages that can switch between lysogenic or lytic life cycles are known as temperate phases. In the lysogenic cycle these bacteriophages are integrated into bacterial genome without disrupting the host cells, where in lytic cycle it becomes a virulent phage.

**Three prime untranslated region (3’ UTR).** The region of mRNA that immediately follows the translation termination codon is known as the 3’ UTR. It often contains regulatory elements (REs) that influence post-transcriptional gene expression.