https://doi.org/10.1093/nar/gkab456

CRISPRloci: comprehensive and accurate annotation of CRISPR–Cas systems

Omer S. Alkhnbashi[1,*,†], Alexander Mitrofanov[1,†], Robson Bonidia[2,†], Martin Raden[1], Van Dinh Tran[1], Florian Eggenhofer[1], Shiraz A. Shah[3], Ekrem Öztürk[1], Victor A. Padilha[2], Danilo S. Sanches[4], André C. P. L. F. de Carvalho[2] and Rolf Backofen[1,5,*]

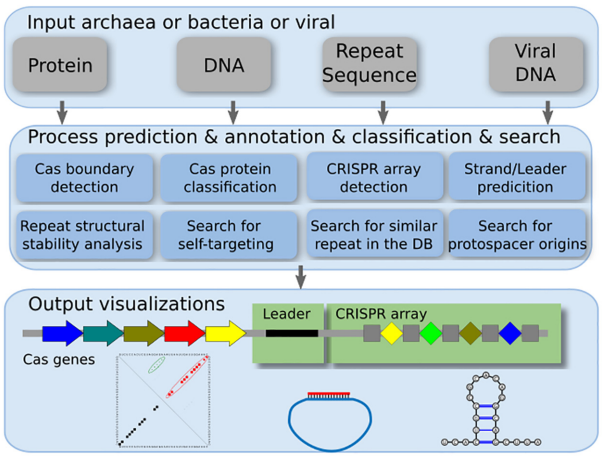
[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Georges-Koehler-Allee 106, 79110 Freiburg, Germany, [2]Institute of Mathematics and Computer Sciences, University of São Paulo, São Carlos, SP, Brazil, [3]Copenhagen Prospective Studies on Asthma in Childhood, Herlev and Gentofte Hospital, University of Copenhagen, Denmark, [4]Universidade Tecnológica Federal do Paraná, Campus Cornélio Procópio, 86300000 Cornélio Procópio, PR, Brazil and [5]Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Schaenzlestr. 18, 79104 Freiburg, Germany

Received March 15, 2021; Revised April 28, 2021; Editorial Decision May 07, 2021; Accepted May 17, 2021

ABSTRACT

CRISPR–Cas systems are adaptive immune systems in prokaryotes, providing resistance against invading viruses and plasmids. The identification of CRISPR loci is currently a non-standardized, ambiguous process, requiring the manual combination of multiple tools, where existing tools detect only parts of the CRISPR-systems, and lack quality control, annotation and assessment capabilities of the detected CRISPR loci. Our CRISPRloci server provides the first resource for the prediction and assessment of all possible CRISPR loci. The server integrates a series of advanced Machine Learning tools within a seamless web interface featuring: (i) prediction of all CRISPR arrays in the correct orientation; (ii) definition of CRISPR leaders for each locus; and (iii) annotation of *cas* genes and their unambiguous classification. As a result, CRISPRloci is able to accurately determine the CRISPR array and associated information, such as: the Cas subtypes; cassette boundaries; accuracy of the repeat structure, orientation and leader sequence; virus-host interactions; self-targeting; as well as the annotation of *cas* genes, all of which have been missing from existing tools. This annotation is presented in an interactive interface, making it easy for scientists to gain an overview of the CRISPR system in their organism of interest. Predictions are also rendered in GFF format, enabling in-depth genome browser inspection. In summary, CRISPRloci constitutes a full suite for CRISPR–Cas system characterization that offers annotation quality previously available only after manual inspection.

GRAPHICAL ABSTRACT



INTRODUCTION

Archaea and bacteria are known to acquire immunity against phages and viruses through a widely conserved RNA-mediated gene silencing pathway (1–3). The non-coding *RNA*s that direct this process are encoded genomically, inside clusters of regularly interspaced short palindromic repeats (CRISPRs). CRISPR loci typically consist of three components: a leader sequence (often AT-rich), followed by an array of direct repeats (DRs), which are separated by so-called spacers, and, optionally, a set of CRISPR-associated *cas* genes. Depending on the type of

*To whom correspondence should be addressed. Tel: +49 761 2037460; Fax: +49 761 2037462; Email: alkhanbo@informatik.uni-freiburg.de
Correspondence may also be addressed to Rolf Backofen. Email: backofen@informatik.uni-freiburg.de
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

CRISPR system, the individual repeats are usually between 18 and 52 nucleotides long (1,4,5). Spacers between the repeats consist of variable sequences, 20 to 50 nt long, which usually match the target genetic elements like phages and viruses (1,6). A CRISPR array, composed of multiple repeats and spacers, can contain anywhere between three to a hundred or more repeat-spacer units, with a total array length exceeding several thousand nucleotides. The leader sequence is between 80 and 500 nucleotides long (7), depending on the type of CRISPR. In general, the leader sequence contains the promoter sequence responsible for generating transcripts of the entire array. The last component is a set of *cas* genes. These genes encode protein complexes that work together with the CRISPR array and its *RNA* transcript to equip the host cell with an adaptive immune system that degrades invading viruses and plasmids (4,8–11). Currently, the CRISPR–Cas systems are classified into two classes: Class 1 systems use multiple Cas proteins that form an interference complex to degrade foreign nucleic acid. This includes types I, III and IV. On the other hand, CRISPR–Cas systems that employ a single interference Cas protein with multiple functional domains for targeting, are assigned to Class 2. This includes type II, V and VI. (5). The fast rate of evolution of CRISPR–Cas systems makes them hard to detect in metagenomic sequences of uncultured bacteria and archaea. Owing to their highly heterogeneous sequences, it is very likely that no counterparts from cultured genomes are similar enough. New Cas proteins usually have a similar three dimensional structure to known Cas proteins, however, their amino acid sequence variability makes the detection challenging, even for the most sophisticated sequence alignment methods (12). While some Cas proteins (for instance, Cas1 and Cas2) are very conserved across different organisms and CRISPR types, other proteins (for instance, Cas7 and Cas8) are too variable to detect similarity even within the same subtype.

In this paper, we propose a new Machine Learning (ML) based webserver, as part of the `Freiburg RNA tools` (13), for identifying, annotating and classifying CRISPR arrays, Cas proteins and Cas cassettes. Besides annotating and classifying CRISPR–Cas systems, `CRISPRloci` can also predict the thermodynamic stability of the CRISPR hairpin motifs within their entire CRISPR arrays and identify self-targeting and phage-host interactions. `CRISPRloci` also provides a host-viral interactions feature by reporting how many spacers potentially originated from an input viral genome.

MATERIALS AND METHODS

Input

`CRISPRloci` offers four different modes of operation, depending on the elements to be annotated. Thus, protein, genomic DNA, CRISPR repeats or viral sequences are accepted (see Figure 1 and Supplementary Table S2 in supplementary materials). The *Genome DNA* mode is the most comprehensive one and screens a prokaryotic genome for CRISPR arrays, determining also their orientation and associated leader sequences. Moreover, it will identify the cassette boundaries, and within these boundaries, the *Cas* proteins together with their subtype classification. There are three sets of parameters available that enable the user to fine-tune the predictions of both CRISPR arrays and *cas* genes. In addition, all the parameters feature tool-tips. The second mode requires a set of prokaryotic protein sequences as input. Our method is sufficiently fast to screen an entire proteome. It identifies and classifies *Cas* proteins, and detects cassette boundaries if protein sequences are provided in the correct order. The third mode accepts one or more CRISPR repeat sequences and identifies the repeat orientation and subtype. Additionally, a search against integrated databases finds regions of local similarity between the input sequences and the list of bona-fide consensus repeats. The fourth mode requires the upload of a complete or partial viral/phage genome. It analyses host-viral connections by reporting how many spacers potentially originated from the input viral genome.

Detection of CRISPR arrays

The task of correctly detecting *CRISPR-array* poses two main difficulties. The first problem lies in the correct identification of the *CRISPR-array* representation, i.e. the array boundaries and the repeat sequence. Once an array-like structure is detected, the second problem is to distinguish a bona-fide *CRISPR-array* from repetitive structures resembling a pseudo CRISPR-array. In our approach we rely on `CRISPRidentify` (14) for both tasks.

To overcome the first challenge, `CRISPRidentify` utilizes consecutive enhancement steps to build multiple candidate representations for each potential *CRISPR-array* region (See Supplementary Table S1 and Figures S2–S8 in Supplementary materials for the comparison with the other tools).

To pick the best representation and simultaneously filter out false candidates, `CRISPRidentify` utilizes a data driven ML based approach. First, it transforms each candidate into a feature vector, where each feature represents a biological property such as repeat length, number of mismatches between repeats, or similarity of spacers etc. Afterwards, the candidate is classified based on the pre-trained ML model. This approach enables the generation of a certainty score for each candidate, and therefore assess the confidence level. After the *CRISPR-array* extraction, the orientation is predicted using `CRISPRstrand` (15). Finally, we enrich the identified array with information about the leader sequence using `CRISPRleader` (6).

It is well known that the secondary structure motif of the CRISPR repeat is essential for the generation and loading of *crRNAs* in many CRISPR–Cas systems. Therefore, after building the set of *CRISPR-arrays*, we analyze the structural stability profiles for the repeats in each *CRISPR-array*. First, we use `RNAfold` (16) to measure the Minimum Free Energy (MFE) of the consensus repeat. Next, we minimize the contribution of long-range base pairs, which are unreliable via a local folding approach for determining base pair probabilities (8). Afterwards, we compute a local structure prediction on the entire *CRISPR-array* using RNAplfold (17) with the windows-size and based-pair-span parameters ($W = 150$ and $L = 80$, respec-

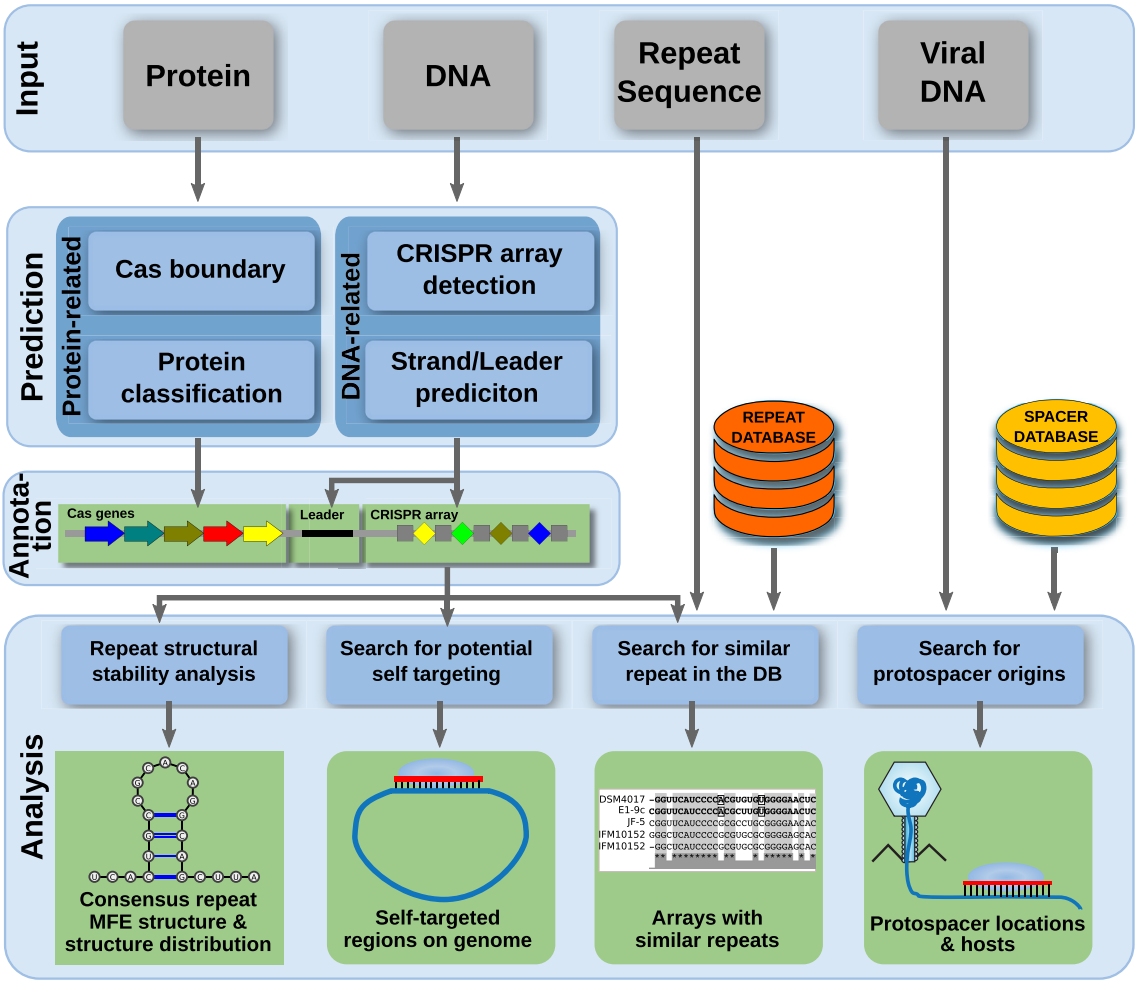


Figure 1. The workflow of `CRISPRloci`. The workflow supports 4 different types of input. If DNA is picked as the input, `CRISPRloci`will identify the *CRISPR arrays*, predict their orientation and the *Leader* sequence and then extract the repeat and spacer sequences. Repeat sequences are then analyzed for their structural stability while spacers are used to identify the potential regions of self targeting. If protein sequences are submitted as input, *CRISPRloci* will classify and report the protein type and role. The user can optionally input a set of repeat sequences. In this scenario, `CRISPRloci`will perform a search of similar repeat sequences in the existing database. The user will be provided with the hits as well as their region, similarity and e-value. Lastly, the user can provide viral DNA as the input. In this scenario, `CRISPRloci`will perform a search for the protospacers using a database of spacers. The user will be provided with the protospacer coordinates as well as the description of the host *CRISPR arrays*.

tively). In addition, we used the option –noLP to disallow lonely base pairs, which usually improves the prediction quality.

Finally, we explore CRISPR self-targeting and alternative functions of CRISPR–Cas systems that extend beyond adaptive immunity. `CRISPRloci` detects the possibilities of the self-targeting spacers in a given genome of interest. To identify positional self-targeting spacers, we extract all spacer sequences from each *CRISPR-array* and scan for exact or partial matches between the spacer and a portion of the genomic sequence that is not part of *CRISPR-arrays*. Furthermore, we classify the context of the match as mobilome or non-mobilome genes to provide information about the possible evolutionary origin.

Boundaries of *Cas* Cassette

In the CRISPR research field, the identification of cassette boundaries plays an essential role in detecting the cassettes, as new unknown *cas* genes must be disentangled from random genes bordering the locus.

We introduced the first tool (named `Casboundary` (18)) that is able to define, based on ML, the cassette boundaries in an automatic manner.

`Casboundary` assumes that the relation between the signature gene (i.e. the main gene used to define a cassette) and any other member of the same cassette is stronger than the relation of the signature gene and any non-member. In particular, we trained two predictive models, using *Extremely Randomized Trees (ERT)* and *Deep Neural Networks (DNN)*, to classify if signature genes and candidate genes belong to the same cassette (*positive* relation) or not (*negative* relation). Given a genome of interest, for each signature gene found on the genome, the tool defines a potential CRISPR region by considering an interval of k genes downstream and k genes upstream to the signature gene (default: $k = 50$). Next, the induced models are employed to predict the label for the relation between the signature

gene and all genes in the potential region. The boundary is specified as the maximal sub-region formed by a list of consecutive genes, such that the first and last gene have positive relations with the signature gene and no more than three consecutive genes with negative relations are permitted.

In the experiments carried out, Casboundary displayed a score of 0.86 for the Jaccard Similarity (JS), which measures the overlap rate between the true and predicted cassettes. On the other hand, CRISPRCasFinder (19), the most similar tool to Casboundary available in the literature, achieved a JS score of 0.70.

Classification of *Cas* proteins and cassette modularization

Considering the high variability of the *Cas* protein sequences, their classification using only standard methods, such as sequence homology or Hidden Markov Models, cannot be easily accomplished. Therefore, we used Casboundary to classify *Cas* proteins according to the known core and signature families. For this task, Casboundary combined features of protein properties with evidence extracted from *Cas* Hidden Markov Models. Based on probabilities that are assigned to a protein to belong to each known *Cas* family, Casboundary was also able to detect proteins that may belong to new putative *Cas* families.

After classifying the *Cas* proteins of the identified cassettes, Casboundary applies a decomposition step that annotates the typical functional modules (adaptation, processing or interference) contained in the cassettes.

Classification of cassettes and prediction of missing proteins

The classification of a cassette subtype is based on the combination of the Cas proteins that it contains (4,11,20). To perform such a task, our CRISPRcasIdentifier tool (21) represents the input cassettes as multidimensional vectors, where each feature corresponds to a different *Cas* protein family, and each value refers to the normalized bit score of each Cas protein family. Thus, we use the different normalized bit scores as evidence that a specific *Cas* protein is contained in a cassette. Next, CRISPRcasIdentifier proceeds to the classification step, which allows the use of three ML algorithms for the induction of classifiers, as follows: CART Decision Tree Algorithm (22), Support Vector Machines (23) and Extremely Randomized Trees (24). During our analysis, we observed that the classifiers correctly identified signatures composed by either one or more genes to determine the cassette subtypes. Such signatures represent the main information that guides the categorization manually performed by experts.

CRISPRcasIdentifier can also predict potentially missing proteins in the input cassettes, based on the remaining proteins. This task is performed by a set of regressors trained to predict the normalized bit scores of each *Cas* family. As a result, it provides evidence for a detailed investigation by the researchers to annotate the missing protein(s).

CRISPRcasIdentifier was compared to five other popular tools from the literature (two webservers and three command-line tools) on the largest public CRISPR benchmark dataset (5). In this analysis, our tool obtained an F-score and balanced accuracy of of 0.91 and 0.89, respectively. On the other hand, the best performances obtained by the other tools were 0.63 and 0.54, respectively.

Virus–host interactions/phage–host interactions

To enhance the study of the mechanisms involved in Virus–Plasmid–Host interactions, it is essential to know the host of a particular virus, phage or plasmid. CRISPRloci therefore provides information for such interactions by detecting all types of matches between a given complete or partial phage genome, and for instance the database of CRISPR spacers from archaeal and bacterial genomes, based on CRISPRidentify.

Processing and implementation

CRISPRloci was implemented with the Freiburg RNA server (13) framework, which is based on Java Server Pages (JSP) processed by an Apache Tomcat server. The jobs of the four different webserver modes are executed within bioconda (25) environments, using pinned tool versions to ensure reproduciblity. The processing time (minutes) for the example datasets provided with the webserver is as follows: 33 (mode 1), 15 (mode 2), 1 (mode 3), 1 (mode 4). For each user submission, a unique link is generated which tracks the progress and retrieves the results upon completion.

RESULTS AND DISCUSSION

CRISPRloci integrates several automated tools to provide comprehensive *in silico* characterization of CRISPR–Cas systems in archaeal and bacterial genomes. Based on complete / draft DNA input, CRISPRloci applies CRISPRidentify to screen for CRISPR arrays (see Figure 1). They are visualized as repeat/spacer pairs. The resulting output (see Figure 2B) features a visualisation of the genomic origin of the array and its subcomponents. This is complemented by rich tabular information (e.g. confidence level indicated by category, consensus repeat sequence, number of repeat sequences and their lengths, the predicted orientation, and the leader region). The dot notation of the repeat representation is used to describe the repeat sequences in detail, indicating conserved regions and mutations with similarity measures and the consensus repeat sequence (see Supplementary Figure S1 in Supplementary materials). Finally, the feature vector shows the corresponding biological properties of the array as well as the certainty score obtained after the feature vector evaluation.

To assess the influence of context on the formation of local structure motifs by the adjacent spacer sequences, CRISPRloci computes the secondary structure stability of each repeat instance in the detected *CRISPR-arrays*, with and without the surrounding sequence context. The dotplot visualisation can provide biological evidence that the sequence context surrounding a repeat might lead to a reduced structure-motif stability. As a result, *crRNA* processing can be inhibited by forming stable base-pairs with repeat sequences that are in conflict with the functional structure motif (see Figure 2A) (8).

To increase both sensitivity and specificity of CRISPR array identification, CRISPRloci builds a set of multiple

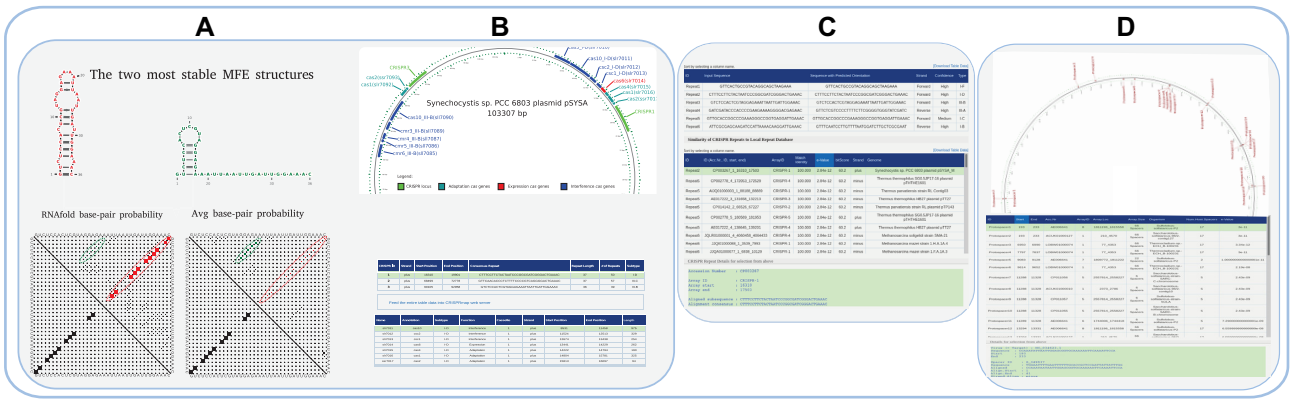


Figure 2. Result data and visualisation. (**A**) In top: `CRISPRloci` shows the two most thermodynamically stable secondary structure candidates, where the minimal free energy structure is highlighted in red. In the bottom: it shows the base-pair probability matrices computed by `RNAfold` and the averaged sub-matrices associated with the repeat structure. Additionally, when we fold the repeat with its sequence context, the corresponding structure is highlighted in green. (**B**) `CRISPRloci` provides a global overview of CRISPR–Cas systems present in the genome and visualizes the results in an interactive genome map and includes the ability to zoom in and click for additional information. (**C**) A table of CRISPR repeat annotation summarizing the results, including strand and subtype. The list is clickable, revealing additional information about the locus of interest, including consensus repeat sequence, array size and organisms that harbour similar CRISPRs. (**D**) An overview of the protospacer sequence locations in the viral/plasmid/phage genome and visualization of the results in an interactive genome map, including a full annotation of spacer sequences in the host genomes.

candidates for a genomic region and utilizes a data driven ML approach to filter out the false candidates. This approach is unique to our tool and demonstrates robust results in filtering out false positive candidates (see supplementary materials).

In summary, `CRISPRloci` has a number of advantages, in terms of array detection and representation from genomic DNA, when compared with other tools designed for CRISPR array detection. First, `CRISPRloci` can form array candidates incorporating insertions and deletions as editing operations for the repeat sequences. Second, our approach can handle complete spacer deletion independently from the location of its occurrence: beginning, middle or end of the array. Finally, our approach can identify the presence of truncated or very damaged repeat sequences in the array (see all mentioned cases in the supplementary materials).

`CRISPRloci` builds on the `Casboundary` tool to analyze protein sequence input (see Figure 1) for the identification of CRISPR cassette boundaries (see Figure 2 B). In the original study, `Casboundary` was applied on the dataset provided by (5), and was shown to effectively identify cassettes with a single module or multiple interference modules. Specifically, it improved the detection of single interference module cassettes by 16% and the detection of multiple interference modules by >60%, when compared to `CRISPRCasFinder`. In addition, for the *Cas* classification task, `Casboundary` was able to detect putative new *Cas* protein types in a study case with previously unseen data (18).

For the classification of cassettes and prediction of potentially missing proteins, `CRISPRloci` employs `CRISPRcasIdentifier` (see Figure 1). During the classification experiments, we noted that the tool was able to correctly model known *Cas* subtype signatures, based only on the *Cas* protein combinations of the cassettes and their subtype labels. In addition, we observed that the ML models relied mainly on the interference module to perform predictions, which agrees with the manual annotations conducted by experts. During the analysis of the prediction of missing proteins, we observed that the induced regressors were able to produce association rules that model sets of *Cas* proteins that tend to co-occur in the same cassette. Observing this, we hypothesized that the prediction of missing *Cas* proteins could help to improve the predictive performance, which was experimentally shown to be true for some combinations of ML and Hidden Markov models.

Regarding CRISPR repeat annotation, `CRISPRloci` predicts the strand of the provided repeats (see Figure 1) and the corresponding subtype along with a confidence level. Additionally, `CRISPRloci` performs a search for CRISPR arrays with similar repeats in the database. Such arrays are then shown to the user along with information about each array's genomic origin, region, orientation and the similarity level with the input CRISPR repeat sequences (see Figure 2 C).

`CRISPRloci` models potential virus-host interactions by identifying protospacer regions from input viral/phage sequences (see Figure 1). Association of the protospacers with their host organisms allows to judge the phylogenetic distribution of the virus targets. Moreover, the output visualizes the location of protospacers mapping to the virus and how many instances of it are present (see Figure 2 D).

As an outlook to the future, we will keep improving the service by adding new functionality, such as tracrRNAs analysis, spacer targeting analysis and improved visualization.

DATA AVAILABILITY

`CRISPRloci` pipeline is implemented in Python, Perl and Java and freely available as both the webserver and standalone versions. The webserver can be accessed via the following link: https://rna.informatik.uni-freiburg.de/CRISPRloci/. The standalone version can be downloaded

from the following GitHub repository: https://github.com/BackofenLab/CRISPRloci.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thanks Sita J. Saunders for the Perl script and Mehmet Tekman for proofreading.

FUNDING

Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy [EXC-2189—Project ID: 390939984, BA 2168/11-1 SPP 1738 and BA 2168/11-2 SPP 1738, BA 2168/3-3, BA 2168/23-1 SPP 2141]; Multiple Functions and Facets of CRISPR–Cas, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) [PROEX-11919694/D]; São Paulo Research Foundation (FAPESP) [2013/07375-0 and 2019/21300-9]; Baden-Wuerttemberg Ministry of Science, Research and Art; University of Freiburg. Funding for open access charge: Baden-Wuerttemberg Ministry of Science, Research and Art and the University of Freiburg.
Conflict of interest statement. None declared.

REFERENCES

1. Barrangou,R. and van der Oost,J. (eds). (2013) In: *CRISPR–Cas Systems: RNA-mediated Adaptive Immunity in Bacteria and Archaea*, Springer Press, Heidelburg, pp. 1–129.
2. Alkhnbashi,O.S., Meier,T., Mitrofanov,A., Backofen,R. and Voss,B. (2020) CRISPR–Cas bioinformatics. *Methods*, **172**, 3–11.
3. Lange,S.J., Alkhnbashi,O.S., Rose,D., Will,S. and Backofen,R. (2013) CRISPRmap: an automated classification of repeat conservation in prokaryotic adaptive immune systems. *Nucleic Acids Res.*, **41**, 8034–8044.
4. Makarova,K.S., Wolf,Y.I., Alkhnbashi,O.S., Costa,F., Shah,S.A., Saunders,S.J., Barrangou,R., Brouns,S. J.J., Charpentier,E., Haft,D.H. *et al.* (2015) An updated evolutionary classification of CRISPR–Cas systems. *Nat. Rev. Microbiol.*, **13**, 722–736.
5. Makarova,K.S., Wolf,Y.I., Iranzo,J., Shmakov,S.A., Alkhnbashi,O.S., Brouns,S.J.J., Charpentier,E., Cheng,D., Haft,D.H., Horvath,P. *et al.* (2020) Evolutionary classification of CRISPR–Cas systems: a burst of class 2 and derived variants. *Nat. Rev. Microbiol.*, **18**, 67–83.
6. Alkhnbashi,O.S., Shah,S.A., Garrett,R.A., Saunders,S.J., Costa,F. and Backofen,R. (2016) Characterizing leader sequences of CRISPR loci. *Bioinformatics*, **32**, i576–i585.
7. Shah,S.A. and Garrett,R.A. (2011) CRISPR/Cas and Cmr modules, mobility and evolution of adaptive immune systems. *Res. Microbiol.*, **162**, 27–38.
8. Reimann,V., Alkhnbashi,O.S., Saunders,S.J., Scholz,I., Hein,S., Backofen,R. and Hess,W.R. (2017) Structural constraints and enzymatic promiscuity in the Cas6-dependent generation of crRNAs. *Nucleic Acids Res.*, **45**, 915–925.
9. Shah,S.A., Erdmann,S., Mojica,F.J.M. and Garrett,R.A. (2013) Protospacer recognition motifs. *RNA Biol.*, **10**, 891–899.
10. Shah,S.A., Alkhnbashi,O.S., Behler,J., Han,W., She,Q., Hess,W.R., Garrett,R.A. and Backofen,R. (2019) Comprehensive search for accessory proteins encoded with archaeal and bacterial type III CRISPR-cas gene cassettes reveals 39 new cas gene families. *RNA Biol.*, **16**, 530–542.
11. Vestergaard,G., Garrett,R.A. and Shah,S.A. (2014) CRISPR adaptive immune systems of Archaea. *RNA Biol.*, **11**, 157–168.
12. Remmert,M. *et al.* (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods*, **9**, 173–175.
13. Raden,M., Ali,S.M., Alkhnbashi,O.S., Busch,A., Costa,F., Davis,J.A., Eggenhofer,F., Gelhausen,R., Georg,J., Heyne,S. *et al.* (2018) Freiburg RNA tools: a central online resource for RNA-focused research and teaching. *Nucleic Acids Res.*, **46**, W25–W29.
14. Mitrofanov,A., Alkhnbashi,O.S., Shmakov,S.A., Makarova,K.S., Koonin,E.V. and Backofen,R. (2020) CRISPRidentify: identification of CRISPR arrays using machine learning approach, *Nucleic Acids Res.*, **49**, e20.
15. Alkhnbashi,O.S., Costa,F., Shah,S.A., Garrett,R.A., Saunders,S.J. and Backofen,R. (2014) CRISPRstrand: predicting repeat orientations to determine the crRNA-encoding strand at CRISPR loci. *Bioinformatics*, **30**, i489–i496.
16. Lorenz,R., Bernhart,S.H., Zu Siederdissen,C.H., Tafer,H., Flamm,C., Stadler,P.F. and Hofacker,I.L. (2011) ViennaRNA Package 2.0. *Algorithm. Mol. Biol.*, **6**, 26.
17. Bernhart,S.H., Mückstein,U. and Hofacker,I.L. (2011) RNA Accessibility in cubic time. *Algorithm. Mol. Biol.*, **6**, 3.
18. Padilha,V.A., Alkhnbashi,O.S., Tran,V.D., Shah,S.A., de Carvalho,A.C.P.L.F. and Backofen,R. (2020) Casboundary: automated definition of integral Cas cassettes. *Bioinformatics*, btaa984.
19. Couvin,D., Bernheim,A., Toffano-Nioche,C., Touchon,M., Michalik,J., Néron,B., Rocha,E.P., Vergnaud,G., Gautheret,D. and Pourcel,C. (2018) CRISPRCasFinder, an update of CRISRFinder, includes a portable version, enhanced performance and integrates search for Cas proteins. *Nucleic Acids Res.*, **46**, W246–W251.
20. Makarova,K.S., Haft,D.H., Barrangou,R., Brouns,S. J.J., Charpentier,E., Horvath,P., Moineau,S., Mojica,F. J.M., Wolf,Y.I., Yakunin,A.F. *et al.* (2011) Evolution and classification of the CRISPR–Cas systems. *Nat. Rev. Microbiol.*, **9**, 467–77.
21. Padilha,V.A., Alkhnbashi,O.S., Shah,S.A., de Carvalho,A.C.P.L.F. and Backofen,R. (2020) CRISPRcasIdentifier: machine learning for accurate identification and classification of CRISPR–Cas systems. *Gigascience*, **9**, https://doi.org/10.1093/gigascience/giaa062.
22. Breiman,L. *et al.* (1984) In: *Classification and Regression Trees*. Chapman & Hall/CRC.
23. Vapnik,V. (1995) In: *The Nature of Statistical Learning Theory*. Springer-Verlag.
24. Geurts,P. *et al.* (2006) Extremely randomized trees. *Mach. Learn.*, **63**, 3–42.
25. Grüning,B., Dale,R., Sjödin,A., Chapman,B.A., Rowe,J., Tomkins-Tinch,C.H., Valieris,R. and Köster,J. (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods*, **15**, 475–476.