Article

DVGfinder: A Metasearch Engine for Identifying Defective Viral Genomes in RNA-Seq Data

María J. Olmo-Uceda [1], Juan C. Muñoz-Sánchez [1], Wilberth Lasso-Giraldo [1], Vicente Arnau [1], Wladimiro Díaz-Villanueva [1], and Santiago F. Elena [1,2,*]

[1] Instituto de Biología Integrativa de Sistemas (I²SysBio), CSIC-Universitat de València, 46980 Paterna, Valencia, Spain
[2] Santa Fe Institute, Santa Fe, NM 87501, USA
[*] Correspondence: santiago.elena@csic.es; Tel.: +34 963 544 779

Abstract: The generation of different types of defective viral genomes (DVG) is an unavoidable consequence of the error-prone replication of RNA viruses. In recent years, a particular class of DVGs, those containing long deletions or genome rearrangements, has gain interest due to their potential therapeutic and biotechnological applications. Identifying such DVGs in high-throughput sequencing data has become an interesting computational problem. Up to nowadays, several algorithms have been proposed, though all incur in false positives, a problem of practical interest if such DVGs have to be synthetized and tested in the laboratory. Here we develop a novel software, DVGfinder, that wraps the two most commonly used algorithms into a pipeline that predicts DVGs. Using a gradient boosting classifier machine learning algorithm, we evaluate the performance of DVGfinder compared to previous algorithms and found that it outcompetes their precision and sensitivity in simulated datasets. DVGfinder generates user-friendly output files in HTML format that can assist users to identify DVGs based on their associated probability of being true positives.

Keywords: benchmarking; bioinformatics; defective viral genomes; gradient boosting; machine learning; RNA-seq; SARS-CoV-2; virus replication

1. Introduction

One of the hallmarks of RNA virus replication is their low fidelity, owed to the lack of proof-reading activities in their RNA-dependent RNA polymerases (RdRp). During viral replication, a plethora of defective copies of the viral genome, or DVGs, can be generated: point mutations and hypermutations, frame-shiftings, short and large deletions, and largely rearranged genomes (*e.g.*, copy- and snap-backs) all form part of the mutant swarm that constitute the viral population. These RNA molecules are called defective genomes because they need the presence of the wild-type viral genome to accomplish their replication cycle [1]. Since non-infectious defective viruses were first described in influenza A virus by Von Magnus and Gard [2], examples of DVGs has been pervasively reported in positive and negative single-stranded RNA viruses [3], doble-stranded RNA viruses and retroviruses [4]. A particular interesting type of DVG is known as defective interfering particles (DIPs), which show the capacity to encapsulate and interfere with the replication and population dynamics of the wild-type virus [5]. In a recent review, DVGs have been considered as long noncoding RNAs, though they might preserve some coding capacity [6].

Recent studies suggest that DVG generation is not a completely stochastic process and could be mediated by viral and host factors [3]. This new observation opens the possibility of manipulating the generation of DVGs (and DIPs) with therapeutic purposes, resulting in the so-called therapeutic interfering particles or TIPs [4]. Indeed, the first TIP rationally designed and synthetically produced to interfere with SARS-CoV-2 has already been published[7].

The first virus-host interaction model with production of DIPs was proposed by Huang and Baltimore [8]. More recent models take into account the immune stimulation effect that DIPs could exhibit [3,5]. As mentioned above, DIPs could be used therapeutically as antivirals [9] and vaccine adjuvants [10], and although the specific mechanisms of generation are not well understood yet, its impact in viral evolutionary dynamics has been well stablished, both experimentally and theoretically [4]. Additional information is clearly needed for better appreciate the implications of DVGs in pathogenesis. For instance, to understand the mechanisms behind their apparently antagonist functions: on the one hand, in most infections DVGs are triggering the host immune system but, in the other hand, evidences exist that involucrate them into viral persistence [11].

The types of DVG frequently considered have been deletions and copy- and snap-backs (Fig. 1) but insertions and rearrangements with host's mRNAs should not be discarded. Although the terminology is not yet completely unified, the DVGs can be characterized by three parameters: the genome position at which the RdRp is released from the template or *break-point* (BP), the genome position at which the RdRp reattaches to the template and continues polymerizing or *reinitation site* (RI) and the sense of the fragments pre- and post-BP/RI junction (Fig. 1).

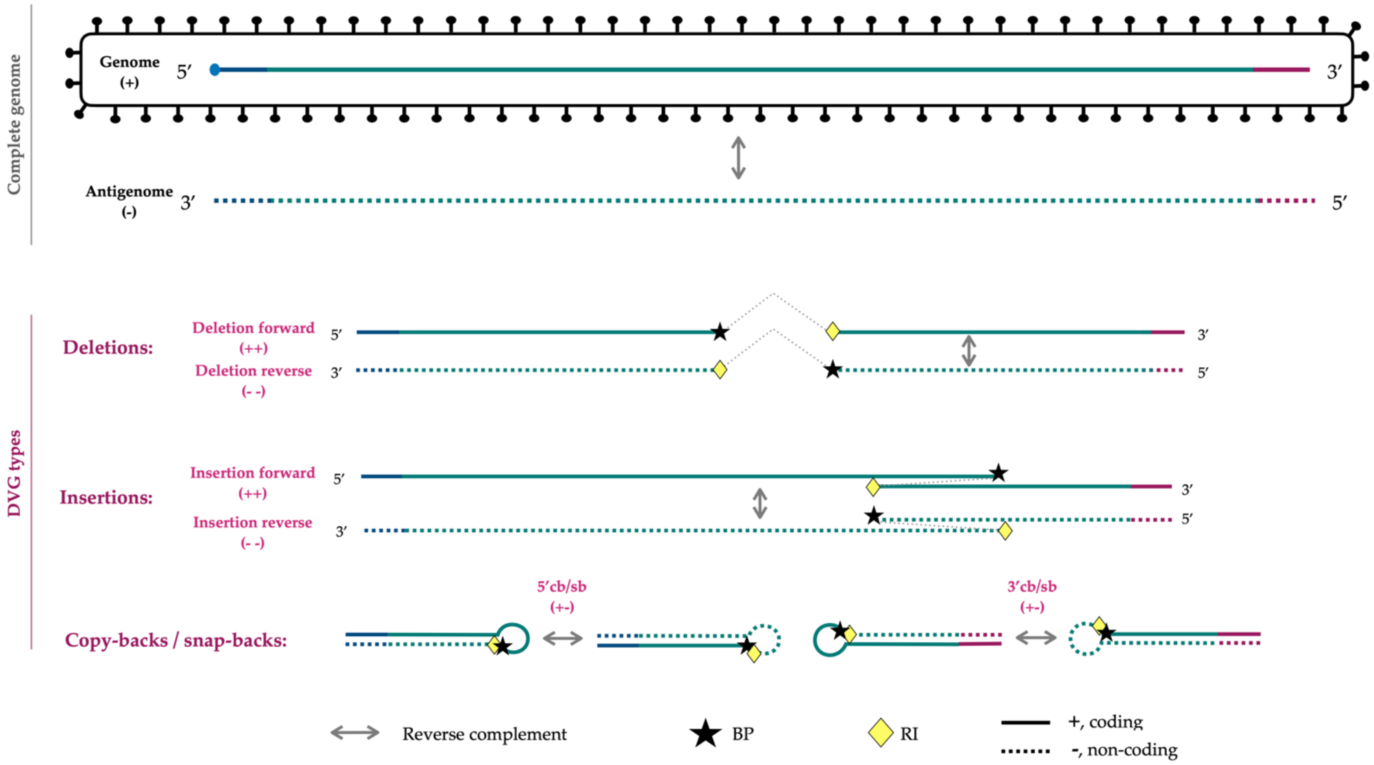

Figure 1. Schematic representation of the different types of DVGs that can be produced during RNA virus replication.

The strategy most often used to identify DVGs in high-throughput sequencing (HTS) data is based on a first step of alignment of all sequenced reads against the reference viral genome and subsequent identification of the BP and RI coordinates. In recent years some programs have been published with this objective. As far as we know, the earlier program that took into account the DVGs was Paparazzi [12] although it was not specifically designed to identify DVGs and excluded them to reconstruct the consensus viral genome in a sample. The first specific program for the DVGs identification and characterization in HTS data was ViReMa-a [13], which together with DI-tector [14] are the most widely used algorithms to date. Table 1 lists the different published bioinformatic tools for DVG detection.

Although a rigorous assessment of the comparative performance of these programs has not been done, is well known that most of these algorithms introduce *in silico* artifacts (*e.g.*, false positives) during its identification and assembly process.

able 1. Published tools for the identification of DVGs. Abbreviatures: D, deletion; I, insertion; cb/sb, copy- and/or snap-back. The last column enumerates the works that have used each program (search on PubMed database, last accessed on 2022/02/23).

Program	Language	DVG type identified	Reference	Number of citations	Citations that have used the program
ViReMa-a	Python	D, I, cb/sb	[13]	31	[15 – 29]
DI-tector	Python	D, I, cb/sb	[14]	9	[25 – 29]
VODKA	Perl	cb/sb	[30]	13	[31]
DVG-profiler	C++	D, I, cb/sb	[27]	4	-
DG-seq	R	D, I, cb/sb	[32]	5	-

Here we present a new machine learning (ML) based package, DVGfinder, that facilitates the identification and extraction of a set of candidate DVGs for further experimental validation. For that, our program integrates the two most commonly used DVG searching tools (Table 1), ViReMa-a and DI-tector, in a single workflow unifying the terminology and adding a set of descriptive variables. Additionally, DVGfinder applies a ML algorithm to reduce the number of false positives and generates a user-friendly and rich HTML report with graphical outputs that help a first exploration of DVG candidates.

2. Materials and Methods

2.1. Implementation, requirements and availability

DVGfinder is written in Python3 and bash and, for now, it runs in the Linux-type command line. It calls the programs ViReMa-a (v.0.23) and DI-tector (v.0.6), the aligners bowtie and bwa and some specific Python libraries. It is highly recommend to create an environment, for that reason, a configuration environment file Conda-like formatted (dvgfinder_env.yaml) with all the external programs and Python libraries requirements is provide in the DVGfinder repository (https://github.com/MJmaolu/DVGfinder).

The parameters needed for execution are -fq, sample in fastq format; -r, virus reference genome in fasta format, also it must exist the indexed genome for bwa and bowtie in the same directory; -m, number of bases to take in account in the measurement of the mean depth over the junction BP-RI (default 5); -t, probability threshold to filter out the true DVGs from false positives (default 0.5); and -n, number of processes to be used.

2.2. Algorithm

The program is structured in three modules acting sequentially (Fig. 2): (1) Metasearch module: launches the search of DVGs in the HTS data with the ViReMa-a and DI-tector algorithms. Then, the program characterizes all the structural detected DVGs and generate informative variables using a unified terminology (see below). (2) Prediction module: assigns to each DVG a probability of being a true positive. (3) Report module: writes final tables, prepares the graphic visualizations, and wraps up everything into an HTML final report.

There are four cases of usage (Fig. 2A) and depending on whether or not the Metasearch module finds candidate DVGs, the Prediction module is launched (Fig. 2A, case 1).

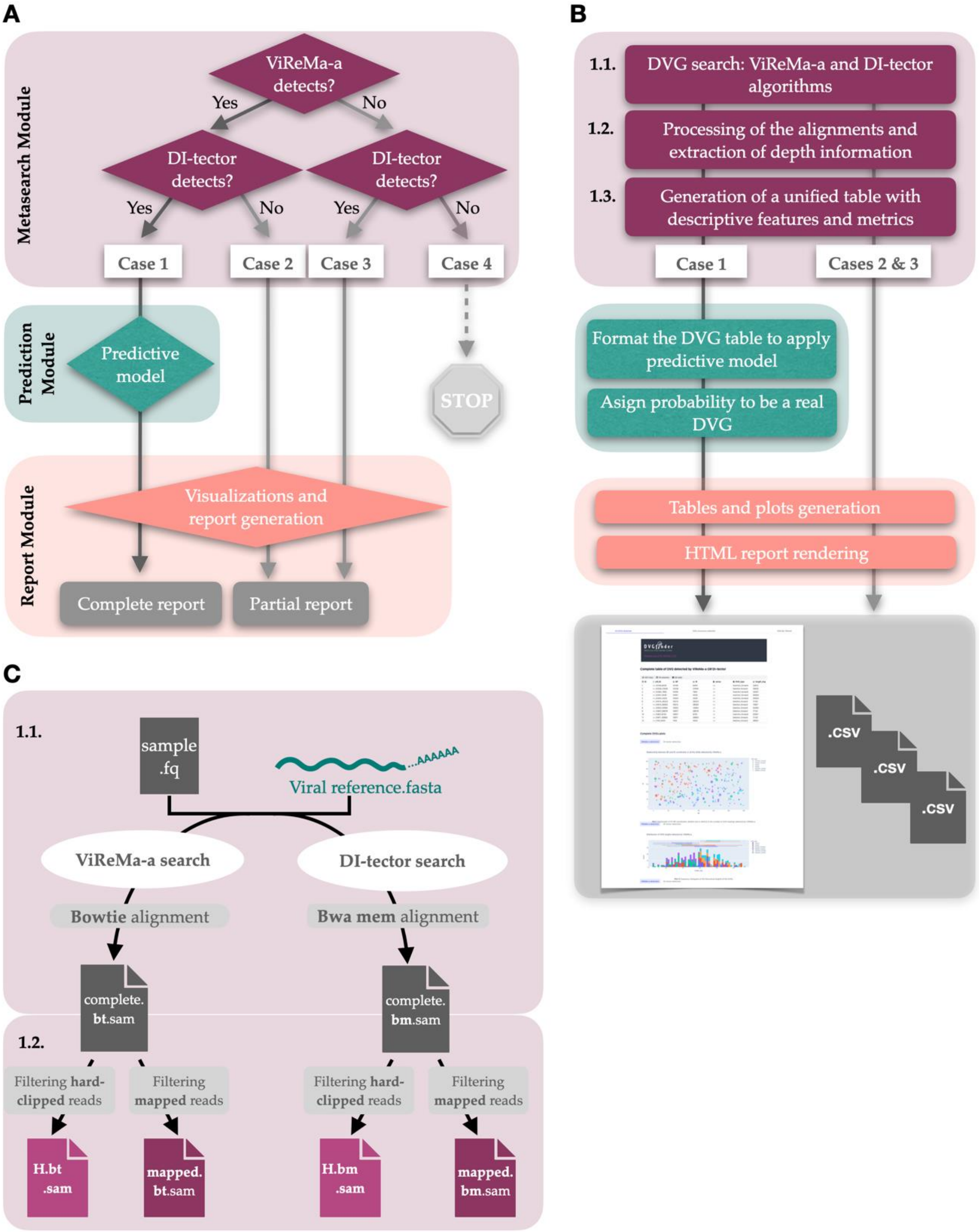

Figure 2. DVGfinder flow chart. (A) General workflow of the program with the four cases of usage accounted, (B) and with a little more of detail. (C) Simplification of the two first steps of the Metasearch module, with emphasis on the processing of the different alignments obtained.

2.3. Descriptive features

The last part of the Metasearch module consists in the addition of descriptive variables to each DVG. These features add information about the depth in both BP and RI coordinates, and about the mean depth in its neighborhoods, both in absolute and normalized values, as explained below. The estimated length of the DVGs is also provided. All this information is saved in a cvs-formated text file ({sample_name}_resume_table.csv). The predictor variables were metrics related to the proportion of the defective genome in the coordinates BP and RI and in its neighborhood as follows:

1. *RPHT* or reads per hundred thousand, relativizes the number of reads with the BP/RI junction to the total number of reads that are correctly align with the reference sequence (mapped reads, first selection (Fig. 2C, 1.2)):
   $$RPHT = 10^5 \times DVG\ read\ counts/total\ mapped\ reads. \quad (1)$$
2. *pBP* and *pRI* or the proportion of reads in the BP and RI genome coordinates, normalizes the number of reads presenting the DVG with respect to the total number of mapped reads at these positions. The values can be greater than one because the search algorithms can find DVGs in reads that do not align with the reference in the first moment.
   $$pX = DVG\ read\ counts/total\ depth\ in\ coordinate\ X, \quad (2)$$
   where $X \in$ [BP, RI].
3. *SDRM* stands for the semi-difference relative to the mean for positions a = depth in BP and b = depth in RI. *SDRM* can also be computed as the depth mean in the neighborhood of the coordinates of interests X and in this case, a = mean pre-coordinate X depth and b = mean post-coordinate X depth:
   $$SDRM = |a - b|/(a + b). \quad (3)$$

These metrics were calculated using the information of the two alignments produced by bowtie and bwa mem, respectively. Also, an additional level of resolution was considered separating the hard-clipped reads from the rest of the mapped reads. Henceforth, the Metasearch module incorporates information about *pX* and *SDRM* from the four alignments, *i.e.*, mapped and hard-clipped from the two alignment algorithms (Fig. 1C).

The final outputs also report characterization variables such as the sense, BP and RI coordinates, the type of DVG in which it has been classified, and the theoretical length of the deletion. In addition, we add a unique identifier of the DVG generated with the format 'sense_BP_RI'. For example, identifier '++_100_300' represents a deletion in coding sense involving the fragment from genomic positions 101 to 299 (Fig. 1). Likewise, identifier '+-_100_80' would represent a copy-back in which nucleotides between positions 81 to 99 form the loop (Fig. 1).

Table 2. Coordinates used for the reconstruction of the theoretical length of the DVGs. The length is calculated as the sum of the nucleotides that conform the pre-BP and post-RI fragments.

DVG type	Fragment	Sense	Coordinates extracted from reference
Deletion forward and insertion forward	pre-BP	+	[start, BP]
	post-RI	+	[RI, end]
Deletion reverse and insertion reverse	pre-BP	-	[BP, end]
	post-RI	-	[start, RI]
5′ cb/sb	pre-BP	+	[start, BP]
	post-RI	-	[start, RI]
3′ cb/sb	pre-BP	-	[BP, end]
	post-RI	+	[RI, end]

2.4. Estimation of the theoretical length of DVG

The program calculates the theoretical length of the DVGs reconstruction. To do this, two reasonable assumptions have been made. Firstly, it has been considered that a DVG only results from an abnormal junction event. Secondly, once the RdRp reattaches in the RI point, it continues the polymerization of the DVG until running off at the end of its template.

2.5. Model generation

A Gradient Boosting Classifier (GBC) was chosen to build the predictive model using the methods implemented by the scikit-learn library [33] and following the DOME recommendations for supervised ML validation in Biology [34]. Before selecting this algorithm, we tested several combinations of algorithms and number of predictors and choose the current one based on its best scores in accuracy, precision, F_1 score, receiver operating characteristic curve (*ROC*), and the area under the ROC curve (*AURC*). Fig. 3 summarizes the process.

2.5.1. Data

The training dataset consists of 1526 events with the categories real and artifact balanced (764 and 762 respectively). This dataset was generated as follows: we created a synthetic fastq file simulating the sequencing of a known set of DVGs (630 different genomes) and its wild type reference genome, using as reference the (+)ssRNA SARS-CoV-2 (NC_045512.2). The Metasearch module of DVGfinder was applied on this sample resulting in a dataset with 1526 candidates that were then labeled as real or artifacts, defining as reals the DVGs directly introduced in the set of known DVG and its reverse complement events. The predictors used had been explained in section 2.3. The labeled dataset is accessible in the DVGfinder's GitHub repository.

2.5.2. Training, optimization and evaluation of the model

The whole synthetic dataset was split into train (75%) and test (25%) sets using the parameter 'stratify = y' from 'train_test_split'.

Eighteen predictors were selected to train the model and through the optimization of the hyperparameters with RandomizedSearchCV function these were adjusted to max_depth = 2, max_features = sqrt, min_samples_leaf = 2, min_samples_split = 4, and n_estimators = 50. The accuracy, recall, precision, F_1 score, and *AURC* of the model were 100-repeated and 4-fold cross-validated in the training set (Fig. 4A) and the optimized model was finally evaluated in the test set (Fig. 4B). The values obtained in the learning and the evaluation process were very close, therefore considering that the model was generalizable. The final model was built with the totally of the data (Fig. 3).

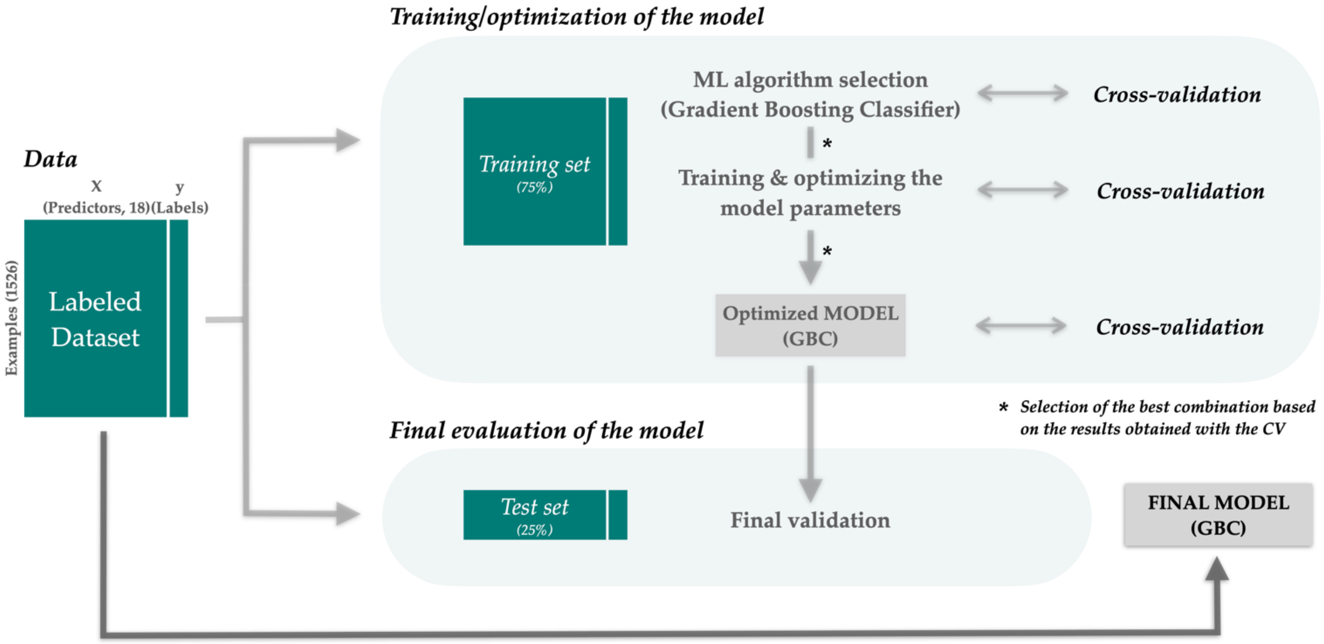

Figure 3. Model generation process. The labeled dataset, composed for a total of 1527 DVGs, between true positives (TP = 764) and false positives (FP = 762) was divided into two separate partitions of data (train and test set) respecting the proportion of the two categories. A GBC was chosen for training the model based on its best performance respect other algorithms such as logistic Regression, Extra Trees Classifier, Supporting Vector Machine, and Random Forest. The optimization of the hyperparameters was achieved by the RandomizedSearchCV function of sklearn library using the training dataset. The optimized model was finally validated on the unseen test dataset considering that the model was generalizable. Finally, the model was trained with the totality of the data and integrated in the Predictive module of DVGfinder.

2.6. Results report

The results are written in different files and a HTML report is generated with the Datapane [35] Python library. In the case 1, the HTML report shows the detected DVGs in three levels: (1) All run mode, with all the DVGs detected by the two search algorithms, (2) Consensus run mode, reporting only the DVGs found by both algorithms, and (3) Filtered run mode, with the DVGs predicted as reals by the ML classifier with a probability greater than the threshold parameter defined by the user (default is 0.5). Furthermore, the three types of graphics generated -scatter plot of BP-RI coordinates, distribution of the DVG length and arc diagram of the deletions- are labeled by the program that had identified them. In cases 2 and 3 the report consists in a one-page HTML with the same data structures as in case 1 but only with the results of the algorithm that found DVGs.

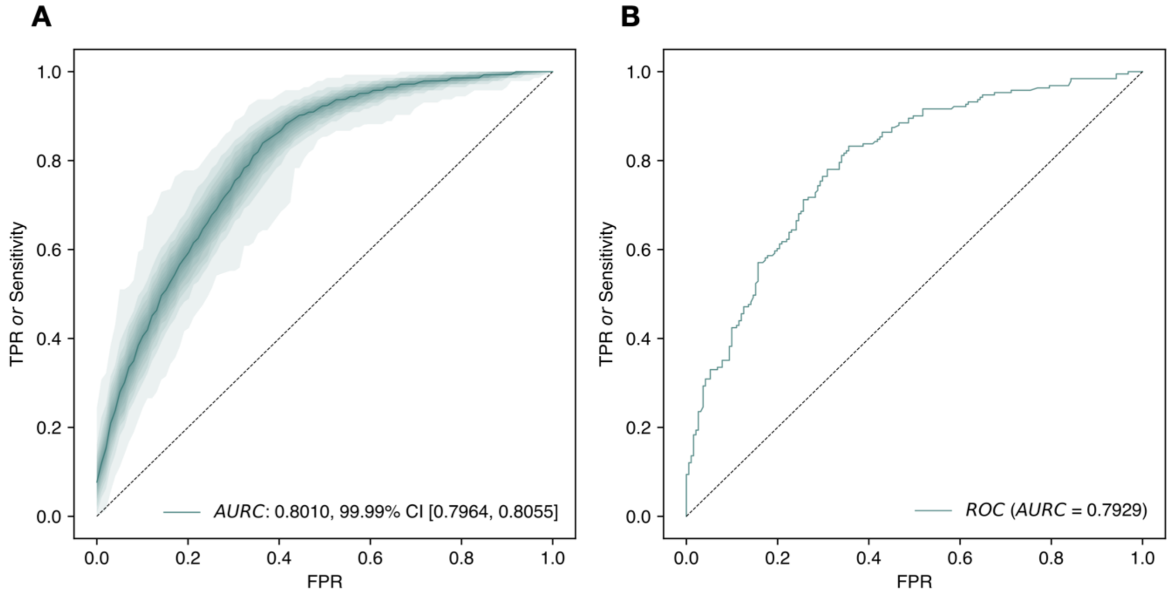

Figure 4. ROC curves of the optimized classifier (GBC algorithm). (A) Four-fold cross validation (100 repeats each fold) in the training set partition, the mean *ROC* is shown with a continuous line and the distribution of the data from 45 - 55 to 0 - 100 percentiles in ranges of 5% in the colored areas (from darker to lighter respectively). A mean *AURC* = 0.8010, with a 99.99% CI from 0.7964 and 0.8055, was achieved in the train set. (B) Final validation of the model in the test partition. Despite these data were completely new for the model, the *AURC* = 0.7929 was really close to the values of the training set, thus considering the model to be generalizable. The final model was trained with the whole data.

2.7. Evaluation data

To assess the performance of the tool we simulated two sets of 12 samples. Both sets were generated with the same DVG density (number of DVGs/genome length) from SARS-CoV-2 and turnip mosaic virus (TuMV) genomes, two (+)ssRNA viruses. The DVGs in the samples were defined randomly, being each type of DVG in equal proportions and supposing the 60% of the total viral population in the samples. The rest of each sample was made up of the reference genome. More precisely, 216 and 72 random DVGs were introduce respectively, and depth samples (*N*) of 10^5, 5×10^5, 10^6, 2×10^6, 3×10^6, 4×10^6, 5×10^6, 6×10^6, 7×10^6, 8×10^6, 9×10^6, and 10^7 of 100 nucleotide-long single end reads were used. In total we used 24 fastq simulated samples.

2.8. Evaluation metrics

The independent results of ViReMa-a and DI-tector and the three output modes of DVGfinder (Metasearch, Consensus and Filtered) were compared on the above synthetic samples. The number of true positives (*TP*), true negatives (*TN*), false positives (*FP*) and false negatives (*FN*) could be quantified in each case. These four quantities were used to compute the following performance indices:
False positive rate $FPR = FP/(FP + TN)$; (5)
precision or positive prediction value $PPV = TP/(FP + TP) = 1 - FDR$; (6)
sensitivity, recall, hit rate, or true positive rate $TPR = TP/(FN + TP)$; and (7)
the harmonic mean of *PPV* and *TPR* or $F_1 = 2 \times PPV \times TPR/(PPV + TPR)$, (8)
where *FDR* in Eq. 6 is the false discovery rate. Also, to facilitate the threshold probability selection $p(TP)$, a study of their impact in *FPR*, *TPR* and precision was performed in both sets of synthetic samples.

2.9. Probability threshold

The prediction model assigns to each candidate DVG a probability of being a true positive ($p(TP)$). By default, DVGfinder shows only the filtered DVGs with $p(TP) > 0.5$. However, we strongly recommend users to adapt this threshold depending on their own confidence requirements. A study of the impact of the $p(TP)$ threshold on *FPR*, *TPR* and *PPV* was performed.

2.10. Statistical evaluation

To assess the differences between the performance of the three DVGfinder run modes and the two searching programs separately, the Friedman rank sum test [36] was used. In the case the null hypothesis was rejected, a pairwise comparison between the DVGfinder run mode with the best median score and each one of the two published search programs, was performed using a Wilcoxon signed rank test with directional alternative to check if the improvement was significant considering all the library sizes. All the *P* values were corrected by Benjamini-Hochberg *FDR* method imposing an overall significance level of 0.05. All the statistical tests were performed with the R software version 4.0.2 in RStudio version 2020.06.22.

3. Results

3.1. Evaluating DVGfinder Performance

We have evaluated the performance of the three DVGfinder run modes (Metasearch, Consensus and Filtered), and the two searching programs (ViReMa-a and DI-tector), separately with the synthetic samples generated from SARS-CoV-2 and TuMV. As the synthetic samples had a known population of DVGs, *TP*, *FP* and *FN* were accounted and sensitivity (*TPR*), precision (*PPV*) and F_1 were calculated for each run mode. The results are shown in Fig. 5. Also, it was tested whether library size had an effect in the difference between the scores and run modes.

Table 3. Comparation of the five modes in the matched samples (twelve for each dataset) using the Friedmann test. In all the scores and datasets has been detected a significant change.

SARS-CoV-2 dataset		
Performance index	**Friedman statistic**	***P***
Sensitivity (*TPR*)	39.5385	5.392×10^{-8}
Precision (*PPV*)	43.8312	6.955×10^{-9}
F_1 score	28.7059	8.970×10^{-6}
TuMV dataset		
Performance index	**Friedman statistic**	***P***
Sensitivity (*TPR*)	33.3214	1.026×10^{-6}
Precision (*PPV*)	39.6123	5.206×10^{-8}
F_1 score	37.6035	1.353×10^{-7}

The results have shown that there are significant differences between the run modes in all the scores and for both datasets (Table 3). In other words, in every multiple comparison exist at least one run mode that performs better than the others.

Pairwise comparisons between the top-scoring DVGfinder run mode and the two search programs are presented and discussed in the following sections.

3.1.1. Sensitivity

The Metasearch run mode shows a higher sensitivity than the search programs ViReMa-a and DI-tector used separately, reaching over 90% of *TPR* since the one million

doi:10.20944/preprints202203.0110.v1

reads samples in both sets of samples (Fig. 5A-B). A Wilcoxon signed rank test for the pairwise comparisons shows that this improvement is statistically significative in the two sets of synthetic data (Table 4).

Table 4. Pairwise comparisons (Wilcoxon signed rank test) for sensitivity in both sets of synthetic samples. The one-tailed contrast performed is shown in the respective column. The *P*-value was corrected with Benjamini-Hochberg *FDR*.

SARS-CoV-2 dataset			
Performance index	**Contrast**	***P***	***FDR***
Sensitivity (*TPR*)	Metasearch - ViReMa-a	0.0012	0.0025
Sensitivity (*TPR*)	Metasearch - DI-tector	0.0012	0.0025
TuMV dataset			
Performance index	**Contrast**	***P***	***FDR***
Sensitivity (*TPR*)	Metasearch - ViReMa-a	0.0011	0.0032
Sensitivity (*TPR*)	Metasearch - DI-tector	0.0111	0.0174

3.1.2. Precision

For the SARS-CoV-2 datasets, the Consensus run mode of DVGfinder shows the best *PPV* in all the library sizes tested (Fig. 5C). For the TuMV dataset, the median of the *PPV* achieved by the Filtered run mode was the same than the median of the Consensus run mode, although the profile was different throughout the library sizes, being dependent on the sample (Fig. 5D). As it is summarized in Table 5, in both sets of samples, the two DVGfinder run modes reaches better precision than DI-tector itself. The improvement respect ViReMa-a and DI-tector used was is significant in both datasets in the case of Consensus run mode, and for the Filtered run mode in TuMV samples too.

Table 5. Pairwise comparisons (Wilcoxon signed rank test) for precision in both sets of synthetic samples. The one-tailed contrast is defined in the contrast column. The *P*-value was corrected with Benjamini-Hochberg *FDR*.

SARS-CoV-2 dataset			
Performance index	**Contrast**	***P***	***FDR***
Precision (*PPV*)	Consensus - ViReMa-a	0.0013	0.0025
Precision (*PPV*)	Consensus - DI-tector	0.0013	0.0025
Precision (*PPV*)	Filtered - ViReMa-a	0.5808	0.6453
Precision (*PPV*)	Filtered - DI-tector	0.0013	0.0025
TuMV dataset			
Performance index	**Contrast**	***P***	***FDR***
Precision (*PPV*)	Consensus - ViReMa-a	0.0173	0.0216
Precision (*PPV*)	Consensus - DI-tector	0.0013	0.0032
Precision (*PPV*)	Filtered - ViReMa-a	0.0122	0.0174
Precision (*PPV*)	Filtered - DI-tector	0.0013	0.0032

An important consideration is that the *PPV* value associated to the Filtered run mode has been calculated using the $p(TP) = 0.5$ default probability threshold, but we highly recommend set up this threshold to reduce the number of *FP*s (and improve *PPV*). This issue will be further discussed in section 3.2.

3.1.3. F_1 score

DVGfinder's Consensus run mode results in the best median F_1 score for both synthetic datasets. However, in some library sizes the Filtered run mode performs better for the TuMV dataset. The contrast between these two DVGfinder run modes and each of the published algorithms was performed as describe in section 2.11. The results in Table 6 show that both Consensus and Filtered run modes significantly improve the F_1 score compared to DI-tector, but not than ViReMa-a.

Table 6. Pairwise comparisons (Wilcoxon signed rank test) for F_1 score in both sets of synthetic samples. The one-tailed contrast is shown in the contrast column. The *P*-value was corrected with Benjamini-Hocheberg *FDR*.

SARS-CoV-2 dataset			
Performance index	**Contrast**	***P***	***FDR***
F_1 score	Consensus - ViReMa-a	0.1277	0.1596
F_1 score	Consensus - DI-tector	0.0027	0.0038
F_1 score	Filtered - ViReMa-a	0.9928	0.9928
F_1 score	Filtered - DI-tector	0.0021	0.0035
TuMV dataset			
Performance index	**Contrast**	***P***	***FDR***
F_1 score	Consensus - ViReMa-a	0.9929	0.9929
F_1 score	Consensus - DI-tector	0.0016	0.0033
F_1 score	Filtered - ViReMa-a	0.9226	0.9929
F_1 score	Filtered - DI-tector	0.0013	0.0032

doi:10.20944/preprints202203.0110.v1

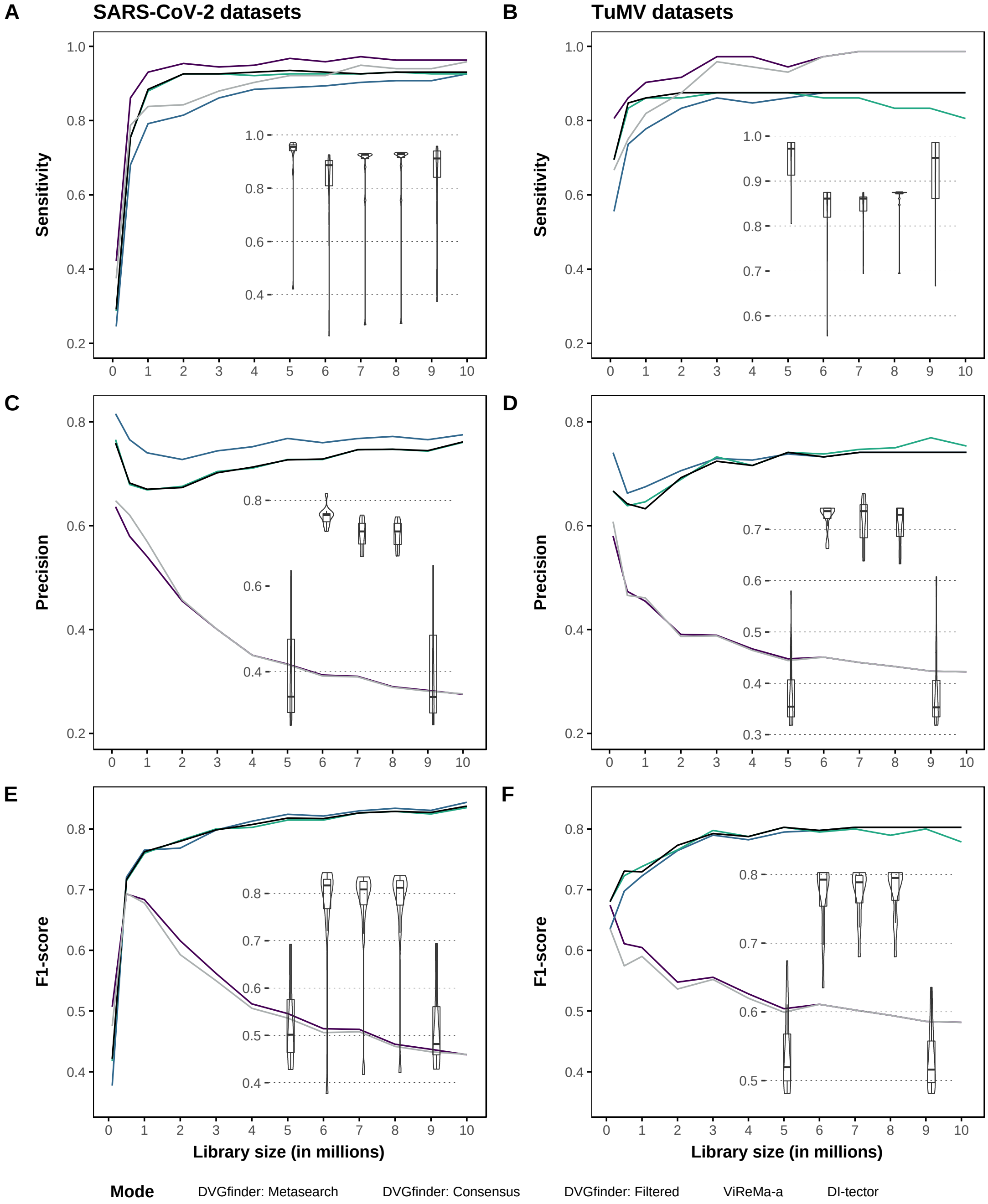

Figure 5. Performance of the three DVGfinder run modes (Metasearch, Consensus and Filtered) and the two search algorithms (ViReMa-a and DI-tector) working separately on the synthetic samples generated from SARS-CoV2 genome (left column) and TuMV genome (right column). Sensitivity (A) and (B), Precision (C) and (D) and F_1 score (E) and (F) were calculated in the twelve synthetic samples generated from each genome. The scores for each run mode are represented in three different types of plots: line graphs show them in relation to the library size while box and rug plots show the distribution of each mode. Within each genome set, each sample contains the same composition in number and type of DVGs, being the size of the library the unique parameter of variation. The library sizes assessed (number of reads) were: 10^5, 5×10^5, 10^6, 2×10^6, 3×10^6, 4×10^6, 5×10^6, 6×10^6, 7×10^6, 8×10^6, 9×10^6, and 10^7.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 March 2022 doi:10.20944/preprints202203.0110.v1

3.2. Exploring the effect of the probability threshold in false positive rate, sensitivity and precision

Using the same synthetic datasets, we performed a study of the effect that the probability threshold $p(TP)$ has in the estimation of FPR, PPV and TPR values using DVGfinder's Filtered run mode. The results from the SARS-CoV-2 and TuMV sets of synthetic samples are shown in Fig. 6. The aim of these representations is to help the user of DVGfinder to better choose the $p(TP)$ argument. Firstly, qualitatively speaking the behavior of the three performance scores is similar for both viruses. Secondly, TPR declines with $p(TP)$ in a non-linear manner, though the decline is larger for TuMV than for SARS-CoV-2 for values $p(TP) < 0.5$. Thirdly, the dependency between precision and threshold probability is not linear either, though it shows a wide range of $p(TP)$ values in which PPV remains quite unaffected (0.3 – 0.7). Interestingly, in terms of reaching a compromise between increasing precision and decreasing TPR, the $p(TP)$ value at which both curves intersect may be a good value choice. For both viruses, this value is in the range 0.65 – 0.7. Furthermore, at this value, the false positive rate is about 0.1 or lower.

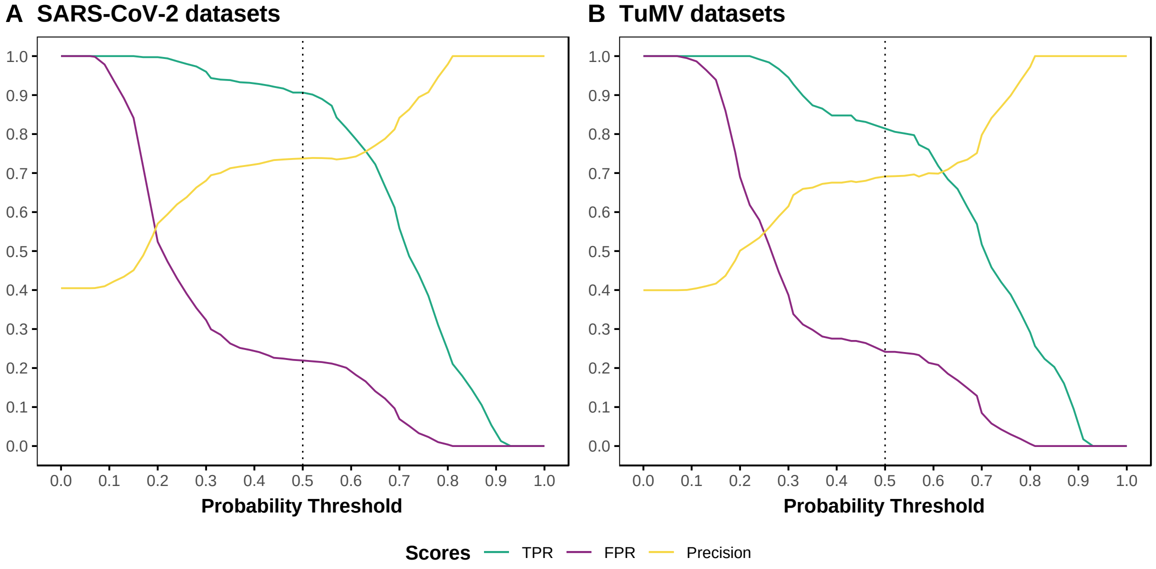

Figure 6. Impact of the probability threshold $p(TP)$ selected in the Filtered run mode of DVGfinder. On each dataset, samples of 0.1 to, 10 million reads library sizes where used. The median values of FPR (Eq. 8), TPR (Eq. 5) and PPV (Eq. 6) are shown with the thick line and the distribution of all the data in intervals of five percentile ranges from 45 - 55% to 0 - 100% in the colored areas (from darker to lighter gradually). (A) Behavior of the probability threshold selection on the SARS-CoV-2 synthetic data and (B) on TuMV synthetic datasets. The dotted line represents the $p(TP) = 0.5$ set as default in the program (this parameter can be adjusted with the argument -t).

3.3. Final report

DVGfinder generates an interactive report with the DVGs identified. An example of the resulting HTML report can be found in http://147.156.206.144/appweb/tumvas72_N100K_l100_report.html. This example shows the results of the TuMV synthetic sample generated as evaluation data, with a library size 100,000 reads.

4. Discussion

There are two published DVG search algorithms that show a high sensitivity but also incur in a number of false positives, ViReMa-a and DI-tector. The strategy we have used in this work consisted in unifying the search process with these two programs, also standardizing the terminology and trying to reduce the number of false positives by applying a ML classificatory algorithm. This is the first time that a machine ML is implemented in a tool for the identification of DVGs in HTS data.

To evaluate the performance of the DVGfinder, we have generated two sets of synthetic samples based on the genomes of two (+)ssRNA viruses, SARS-CoV-2 and TuMV, that widely differ in their genome length and gene content and expression. Each synthetic dataset was composed of a known and equal number of DVGs that included deletions, insertions and copy/snap-backs. The introduced DVGs were randomly selected, constituting the 60% of the total sample reads; the rest of the synthetic sample was composed of the native viral genome. All reads were 100 nucleotides long and libraries of between 10^5 and 10^7 reads were generated. In total, 24 simulated fastq samples were used to assess the sensitivity, precision and the F_1 score of the three DVGfinder run modes as well as the two search programs separately.

These evaluations have shown that DVGfinder's Metasearch run mode significantly improved the sensitivity of both search algorithms used separately. This behavior was expected since the Metasearch run mode consists of the union of the DVGs identified by each algorithm. Although it was not the purpose of this study, we observed that while ViReMa-a achieves better sensitivity in SARS-CoV-2 samples, it is DI-tector that scores better in the TuMV samples. TuMV genome is approximately three times shorter than that of the SARS-CoV-2, suggesting that the algorithm implemented in DI-tector may work better on shorter and less complex genomes. Although further studies of the behavior of the search algorithms on different types of viral genomes is needed, we conclude that DVGfinder, as a metasearch engine, facilitates the benchmarking of the tools that integrate them. In addition, DVGfinder architecture would easily allow to incorporate search algorithms that might be published in the future.

The Consensus run mode of DVGfinder provides a significant improvement in precision respect the two published programs in both groups of datasets. The Filtered run mode, significantly improves the precision of DI-tector in both datasets. However, this run mode only improves the precision reached by ViReMa-a in the case of TuMV synthetic samples. Interestingly, the precision of the Filtered run mode improves drastically by using a threshold > 0.5. This behavior provides a tip to potential users to tune the -t parameter when setting their DVGfinder runs. We recommend choosing a probability threshold around 0.7, as the model shows higher precision and a considerable sensitivity, but obviously this depends on what the user needs to prioritize.

Although the prediction model was trained with a dataset generated from SARS-CoV-2 genome, it works with other (+)ssRNA viral genomes, as TuMV. This could be because the features selected as predictors are not genome-dependent but instead depth-dependent. *I.e.*, the metrics computed with the software and used as input for the ML classifier are related to the depth of the samples at the BP and RI coordinates and surrounding sequences.

Regarding the F_1 score, that weights both sensitivity and precision values, we observed a significance improvement of Consensus and Filtered run modes over DI-tector in both synthetic datasets.

An important aspect to keep in mind is that the search programs that integrate DVGfinder, identify the recombination events present in a sample, but do not provide information about the potential role of these DVGs as defective interfering particles. To facilitate this study, a theoretical DVG length is calculated but we strongly recommend carefully considering the assumptions from which these reconstructions were done. In the one hand, DVGfinder takes the recombination event as the unique abnormal junction in the DVGs. In the other hand, it assumes that once the RdRp reattaches to the template, it continues polymerizing until running off at its extreme. These two simplifications could generate structures larger than the wild type genome length (in some cases, even twice its size), the type of structures that could be considered as artifacts.

Lastly, ViReMa-a output groups identified recombination events by the sense of the sequences before and after the recombination point. By contrast DI-tector output groups possible recombinants by DVG type. We have homogenized the output and proposed a unique identifier with the form 'sense_BP_RI'. This identifier provides into a single tag all the relevant structural information about any DVG.

5. Conclusions

We present DVGfinder, a friendly and easy-to-use metasearch tool to identify DVGs in RNA-seq data. The program integrates the two DVG search algorithms most cited along the last years, ViReMa-a and DI-tector, in a simple workflow. Also, a ML model was trained to sieve true positives from the total number of events identified by the program. The results of the program are written in various csv files with a unified nomenclature. In addition, an interactive HTML report with summary tables and different plots is generated to facilitate an easy first exploration of the data. We propose 'sense_BP_RI' as a unique identifier of the DVGs.

As long as the two search algorithms detect DVGs in the sample, the results are grouped in three run modes: Metasearch mode shows the join of the events detected, Consensus mode selects the intersected events and Filtered mode only the DVGs with a probability equal or higher than a user-defined threshold probability of true positives. In any case, the user can confirm in the output HTML which algorithm has identified the DVG.

We tested the performance of each run mode separately in two sets of synthetic samples based on SARS-CoV-2 and TuMV genomes. The results show that the Metasearch run mode of DVGfinder has a significant improvement in sensitivity, while Consensus and Filtered run modes are more precise than the search programs used separately. The Consensus and Filtered run modes also showed a significant improvement in F_1 score than DI-tector used on its own. We could verify that although the ML classifier was trained on a SARS-CoV-2 genome-based dataset, it also performs well with samples from other (+)ssRNA viruses as structurally different as TuMV.

Author Contributions: M.J.O.-U. implemented the Python code, performed the analyses, and wrote the draft. S.F.E. conceptualized and supervised the study and finalized the manuscript. J.C.M.-S., W.L.-G., V.A., and W.D.-V. provided substantial discussion and guidance. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Generalitat Valenciana grant PROMETEO2019/012, Spain Agencia Estatal de Investigación-FEDER grant PID2019-103998GB-I00 and European Commission – NextGenerationEU (Regulation EU 2020/2094) through CSIC's Global Health Platform (PTI Salud Global) grant SGL2021-03-009. M.J.O.-U. was supported by grant FPU19/05246 from Spain Ministry of Universities.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All scripts developed in this work are available at https://github.com/MJmaolu/DVGfinder, last accessed 10 January 2022.

Acknowledgments: We acknowledge Julia Hillung and Asher Leeks for enlightening discussion on SARS-CoV-2 DVGs, Andrés Bravo for helpful discussions on statistical issues and Juanjo Martínez for creating the DVGfinder logos.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Rezelj, V.V.; Carrau, L., Merwaiss, F.; Levi, L.I.; Erazo, D.; Tran, Q.D.; Henrion-Lacritick, A.; Gausson, V.; Suzuki, Y.; Shengjuler, D.; *et al*. Defective viral genomes as therapeutic interfering particles against flavivirus infection in mammalian and mosquito hosts. *Nat. Commun*. **2021**, *12*, 2290.

[2] Von Magnus, P.; Gard, S. Studies on interference in experimental influenza. II. Purification and centrifugation experiments. *Ark. Kem. Mineral. Geol*. **1947**, *8*, 4.

[3] Ziegler, C.M.; Botten, J.W. Defective interfering particles of negative-strand RNA viruses. *Trends Microbiol*. **2020**, *28*, 554–565.

[4] Vignuzzi, M.; López, C.B. Defective viral genomes are key drivers of the virus–host interaction. *Nat. Microbiol*. **2019**, *4*, 1075–1087, 2019.

[5] Genoyer, E.; López, C.B. The impact of defective viruses on infection and immunity. *Annu. Rev. Virol*. **2019**, *6*, 547–566, 2019.

[6] Shrestha, N.; Bujarski, J.J. Long noncoding RNAs in plant viroids and viruses: A review. *Pathogens* **2020**, *9*, 765.

[7] Chaturvedi, S.; Vasen, G.; Pablo, M.; Chen, X.; Beutler, N.; Kumar, A.; Tanner, E.; Illouz, S.; Rahgoshay, D.; Burnett, J.; *et al*. Identification of a therapeutic interfering particle—A single-dose SARS-CoV-2 antiviral intervention with a high barrier to resistance. *Cell* **2021**, *184*, 6022-6036.e18.

[8] Huang, A.S.; Baltimore, D. Defective viral particles and viral disease processes. *Nature* **1970**, *226*, 325–327.

[9] Yang, Y., Lyu, T.; Zhou, R.; He, X.; Ye, K.; Xie, Q.; Zhu, L.; Chen, T.; Shen, C.; Wu, Q.; *et al*. The antiviral and antitumor effects of defective interfering particles/genomes and their mechanisms. *Front. Microbiol*. **2019**, *10*, 1852.

[10] Wignall-Fleming, E.B.; Vasou, A.; Young, D.; Short, J.A.L.; Hughes, D.J. Goodbourn, S.; Randall, R.E. Innate intracellular antiviral responses restrict the amplification of defective virus genomes of parainfluenza virus 5. *J. Virol*. **2020**, *94*, e00246-20.

[11] Xu, J.; Sun, Y.; Li, Y.; Ruthel, G.; Weiss, S.R.; Raj, A.; Beiting, D.; López, C.B. Replication defective viral genomes exploit a cellular pro-survival mechanism to establish paramyxovirus persistence. *Nat. Commun*. **2017**, *8*, 799.

[12] Vodovar, N.; Goic, B.; Blanc, H.; Saleh, M.C. *In silico* reconstruction of viral genomes from small RNAs improves virus-derived small interfering RNA profiling. *J. Virol*. **2011**, *85*, 11016–11021.

[13] Routh, A.; Johnson, J.E. Discovery of functional genomic motifs in viruses with ViReMa-a virus recombination mapper-for analysis of next-generation sequencing data. *Nucleic Acids Res*. **2014**, *42*, e11.

[14] Beauclair, G.; Mura, M.; Combredet, C.; Tangy, F.; Jouvenet, N.; Komarova, A.V. DI-tector: Defective interfering viral genomes' detector for next-generation sequencing data. *RNA* **2018**, *24*, 1285–1296.

[15] Gribble, J.; Stevens, L.J.; Agostini, M.L.; Anderson-Daniels, J.; Chappell, J.D.; Lu, X.; Pruijssers, A.J., Routh, A.L.; Denison, M.R. The coronavirus proofreading exoribonuclease mediates extensive viral recombination. *PLoS Pathog*. **2021**, *17*, e1009226.

[16] Muruato, A.; Vu, M.N.; Johnson, B.A.; Davis-Gardner, M.E.; Vanderheiden, A.; Lokugamage, K.; Schindewolf, C.; Crocquet-Valdes, P.A., Langsjoen, R.M.; Plante, J.A.; *et al*. Mouse Adapted SARS-CoV-2 protects animals from lethal SARS-CoV challenge. *PLoS Biol*. **2021**, *19*, e3001284.

[17] Jaworski, E.; Langsjoen, R.M.; Mitchell, B.; Judy, B.; Newman, P.; Plante, J.A.; Plante, K.S.; Miller, A.L.; Zhou, Y.; Swetnam, D.; *et al*. Tiled-ClickSeq for targeted sequencing of complete coronavirus genomes with simultaneous capture of RNA recombination and minority variants. *eLife* **2021**, *10*, e68479.

[18] Nilsson-Payant, B.E.; Blanco-Melo, D.; Uhl, S.; Escudero-Pérez, B.; Olschewski, S.; Thibault, P.; Panis, M.; Rosenthal, M.; Muñoz-Fontela, C.; Lee, B. *et al*. Reduced nucleoprotein availability impairs negative-sense RNA virus replication and promotes host recognition. *J. Virol*. **2021**, *95*, e02274-20.

[19] Smith, S.C.; Gribble, J.; Diller, J.R.; Wiebe, M.A.; Thoner Jr, T.W.; Denison, M.R.; Ogden, K.M. Reovirus RNA

recombination is sequence directed and generates internally deleted defective genome segments during passage. *J. Virol*. **2021**, *95*, e02181-20.

[20] Langsjoen, R.M.; Muruato, A.E.; Kunkel, S.R.; Jaworski, E.; Routh, A. Differential alphavirus defective RNA diversity between intracellular and extracellular compartments is driven by subgenomic recombination events. *mBio* **2020**, *11*, e00731-20.

[21] Kautz, T.F.; Jaworski, E.; Routh, A.; Forrester, N.L. A low fidelity virus shows increased recombination during the removal of an alphavirus reporter gene. *Viruses* **2020**, *12*, 660.

[22] Alnaji, F.G.; Holmes, J.R.; Rendon, G.; Vera, J.C.; Fields, C.J.; Martin, B.E.; Brooke, C.B. Sequencing framework for the sensitive detection and precise mapping of defective interfering particle-associated deletions across influenza A and B viruses. *J. Virol*. **2019**, *93*, e00354-19.

[23] Jaworski, E.; Routh, A. Parallel ClickSeq and Nanopore sequencing elucidates the rapid evolution of defective-interfering RNAs in Flock House virus. *PLoS Pathog*. **2017**, *13*, e1006365.

[24] Xu, C.; Sun, X.; Taylor, A.; Jiao, C.; Xu, Y.; Cai, X.; Wang, X.; Ge, C.; Pan, G.; Wang, Q.; *et al*. Diversity, distribution, and evolution of tomato viruses in China uncovered by small RNA sequencing. *J. Virol*. **2017**, *91*, e00173-17.

[25] Bifani, A.M.; Choy, M.M.; Tan, H.C.; Ooi, E.E. Attenuated dengue viruses are genetically more diverse than their respective wild-type parents. *NPJ Vaccines* **2021**, *6*, 76.

[26] Bosworth, A.; Rickett, N.Y.; Dong, X.; Ng, L.F.P.; García-Dorival, I.; Matthews, D.A.; Fletcher, T.; Jacobs, M.; Thomson, E.C.; *et al*. Analysis of an Ebola virus disease survivor whose host and viral markers were predictive of death indicates the effectiveness of medical countermeasures and supportive care. *Genome Med*. **2021**, *13*, 5.

[27] Bosma, T.J.; Karagiannis, K.; Santana-Quintero, L.; Ilyushina, N.; Zagarodnyaya, T.; Petrovskaya, S.; Laassri, M.; Donnelly, R.P.; Rubin, S.; Simonyan, V.; *et al*. Identification and quantification of defective virus genomes in high throughput sequencing data using DVG-profiler, a novel post-sequence alignment processing algorithm. *PLoS ONE* **2019**, *14*, e0216944.

[28] Johnson, R.I.; Boczkowska, B.; Alfson, K.; Weary, T.; Menzie, H.; Delgado, J.; Rodriguez, G.; Carrion Jr, R.; Griffiths, A. Identification and characterization of defective viral genomes in Ebola virus-infected rhesus macaques. *J. Virol*. **2021**, *95*, e00714-21.

[29] Addetia, A.; Phung, Q.; Bradley, B.T.; Lin, M.J.; Zhu, H.; Xie, H.; Huang, M.L.; Greninger, A.L. *In vivo* generation of BK and JC polyomavirus defective viral genomes in human urine samples associated with higher viral loads. *J. Virol*. **2021**, *95*, e00250-21.

[30] Sun, Y.; Kim, E.J.; Felt, S.A.; Taylor, L.J.; Agarwal, D.; Grant, G.R.; López, C.B. A specific sequence in the genome of respiratory syncytial virus regulates the generation of copy-back defective viral genomes. *PLoS Pathog*. **2019**, *15*, e1008099.

[31] Felt, S.A.; Sun, Y.; Jozwik, A.; Paras, A.; Habibi, M.X.; Nickle, D.; Anderson, L.; Achouri, E.; Feemster, K.A.; Cárdenas, A.M.; *et al*. Detection of respiratory syncytial virus defective genomes in nasal secretions is associated with distinct clinical outcomes. *Nat. Microbiol*. **2021**, *6*, 672–681.

[32] Boussier, J.; Munier, S.; Achouri, E.; Meyer, B.; Crescenzo-Chaigne, B.; Behillil, S.; Enouf, V.; Vignuzzi, M.; van der Werf, S.; Naffakh, N. RNA-seq accuracy and reproducibility for the mapping and quantification of influenza defective viral genomes. *RNA* **2020**, *26*, 1905–1918.

[33] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; *et al*. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res*. **2011**, *12*, 2825–2830.

[34] Walsh, I.; Fishman, D.; Garcia-Gasulla, D.; Titma, T.; Pollastri, G.; ELIXIR Machine Learning Focus Group; Harrow, J.; Psomopoulos, F.E.; Tosatto, S.C.E. DOME: recommendations for supervised machine learning validation in biology. *Nat. Methods* **2021**, *18*, 1122-1127.

[35] Datapane. Available online: https://github.com/datapane/datapane (accessed 29 12 2021).

[36] Conover, W.J. *Practical nonparametric statistics*. 3rd ed.; John Wiley: New York, USA, 1999; pp.