VirHunter – my PhD journey of virus detection using machine learning

Sukhorukov Grigorii
under the supervision of Macha Nikolski

université de BORDEAUX
ibgc
cnrs
cbib BIOINFORMATIQUE
INEXTVIR

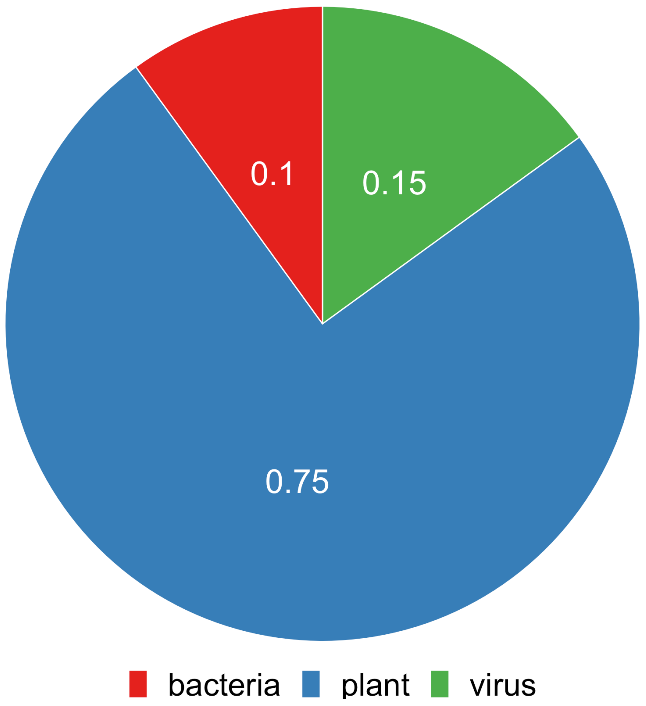
Table of contents

- VirHunter
  - Backgound and context: why we developed VirHunter
  - What is under the hood
  - Potential of novel virus detection with VirHunter
  - Practical aspects of VirHunter
- Decontaminator
- Viroidcatcher

Background and context

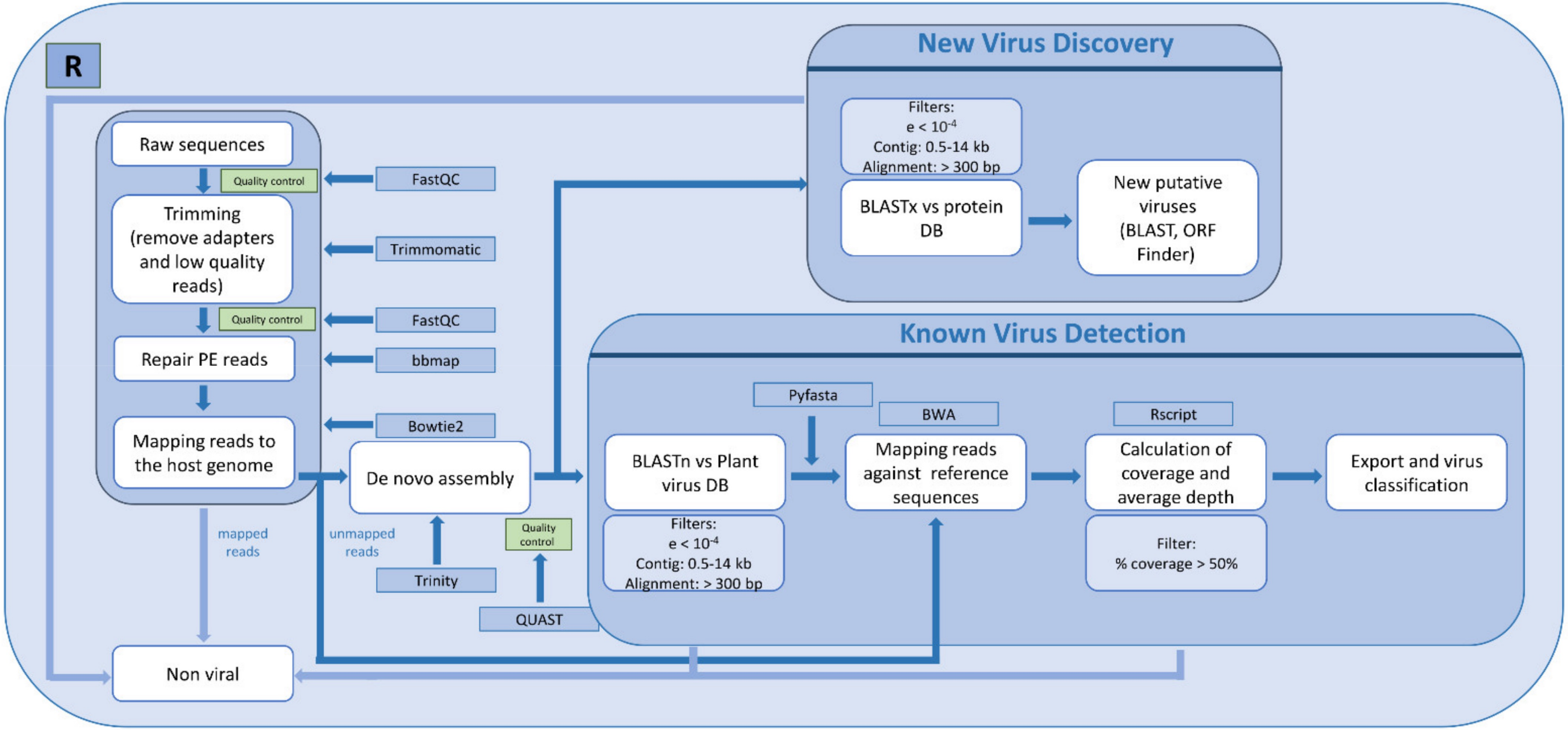
Virus detection in RNAseq data

Example of RNAseq sample read content



- No universal marker genes for RNA viruses
- RNA viruses are highly variable
- RNA viruses from an RNAseq sample often do not have full assembly
- Knowledge in databases is incomplete

Typical workflow for virus discovery



Credits for the image: Ayoub Maachi

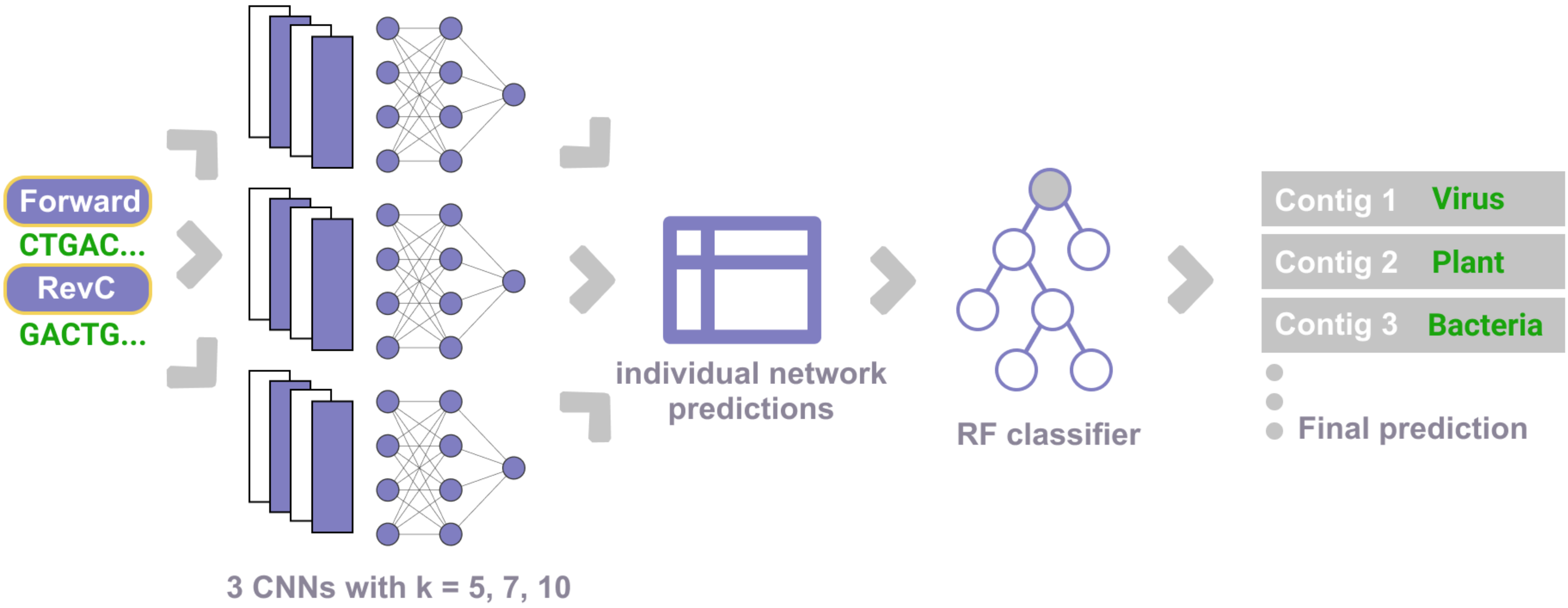
Time-consuming both computationally and in terms of expert analysis

Possible solution: VirHunter

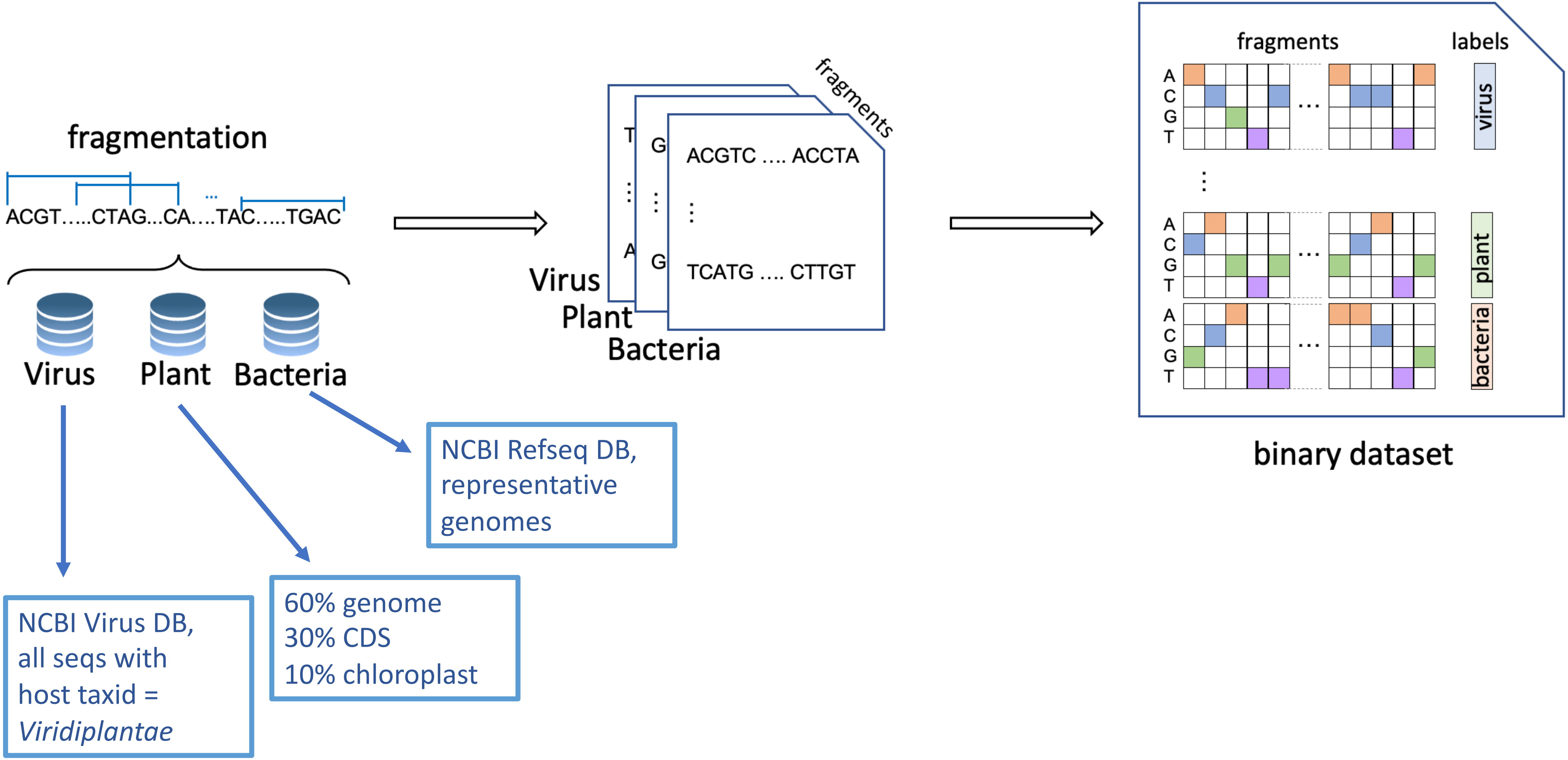
- Works with assembled contigs from plant virome RNAseq samples
- Classifies contigs into *viral*, *plant* and *bacterial* categories
- Is fast and accurate

What is under the hood

Global VirHunter architecture



Learning from annotated data



NCBI Virus DB, all seqs with host taxid = *Viridiplantae*

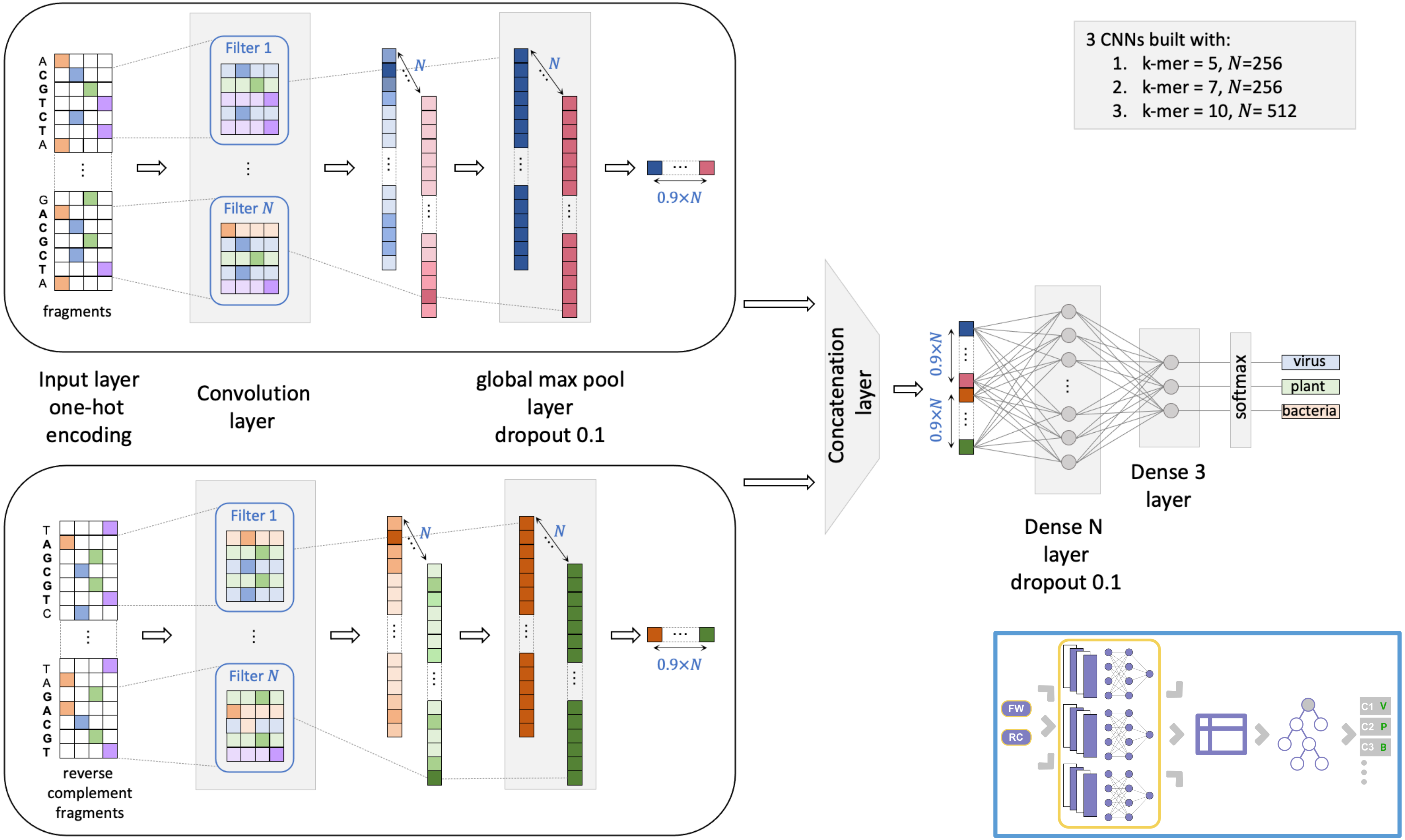
60% genome
30% CDS
10% chloroplast

NCBI Refseq DB, representative genomes

VirHunter Neural Network component

3 CNNs built with:

1. k-mer = 5, N=256
2. k-mer = 7, N=256
3. k-mer = 10, N= 512



Input layer one-hot encoding

Convolution layer

global max pool layer dropout 0.1

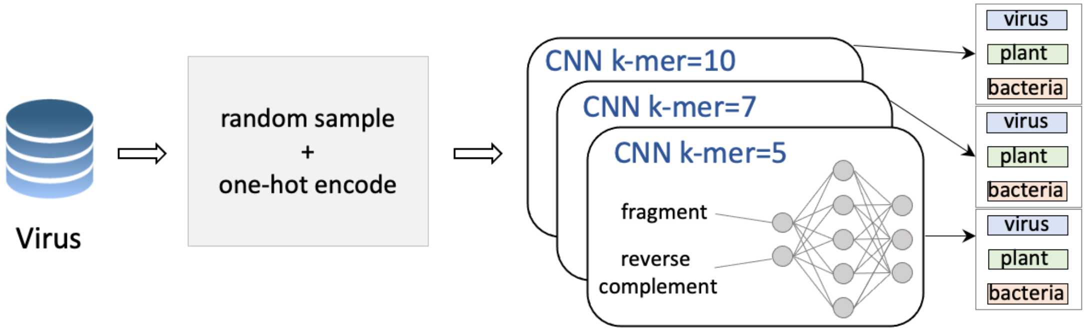
Concatenation layer

Dense N layer dropout 0.1

Dense 3 layer

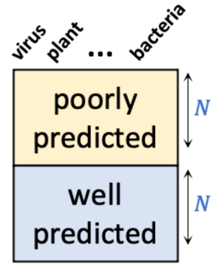
softmax

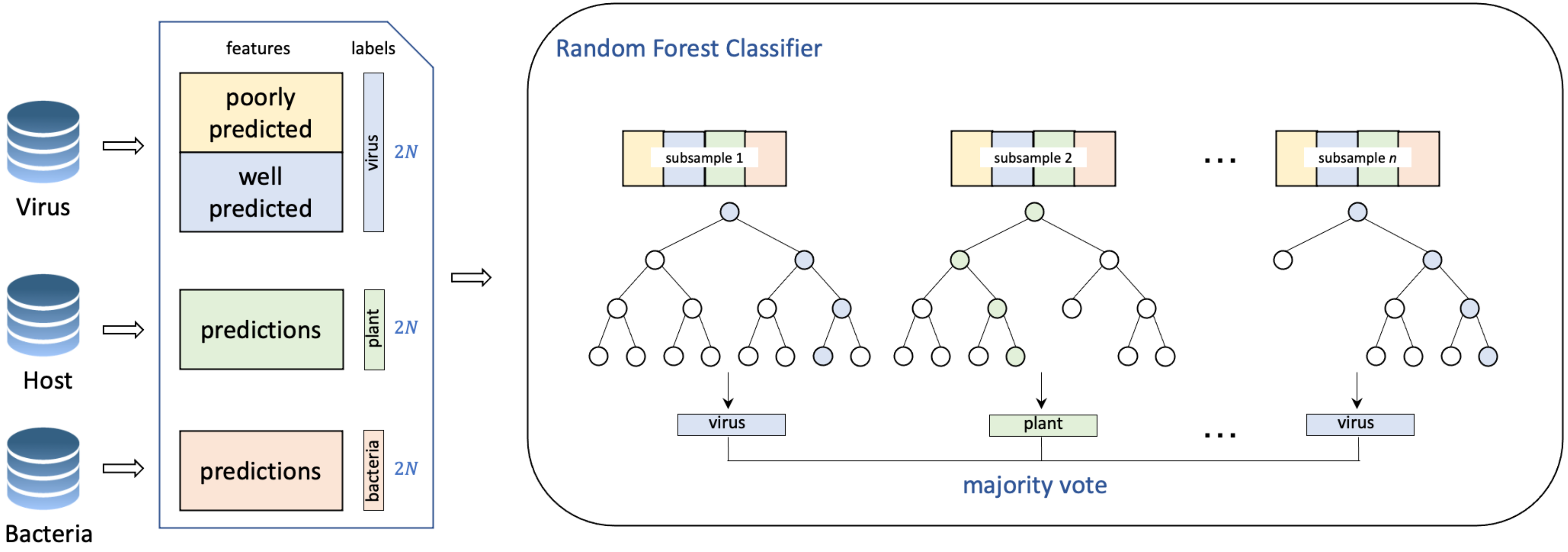
VirHunter Random Forest component

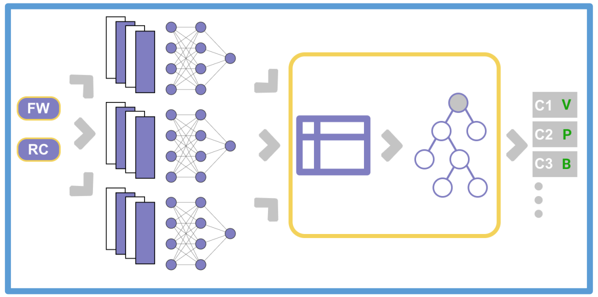


=

CNN k-mer=5			CNN k-mer=7			CNN k-mer=10		
virus	plant	bacteria	virus	plant	bacteria	virus	plant	bacteria
0.8	0.1	0.1	0.5	0.3	0.2	0.9	0.0	0.1
0.2	0.7	0.1	**0.4**	0.5	0.1	**0.6**	0.2	0.2
0.7	0.1	0.2	0.5	0.3	0.2	0.7	0.2	0.1
0.6	0.1	0.1	0.7	0.3	0.0	0.7	0.2	0.1
0.3	0.1	0.6	**0.2**	0.2	0.6	**0.7**	0.2	0.1
0.8	0.1	0.1	0.5	0.3	0.2	0.9	0.0	0.1
0.2	0.7	0.1	**0.4**	0.4	0.2	**0.1**	0.8	0.1
0.7	0.1	0.2	0.5	0.3	0.2	0.7	0.2	0.1
0.6	0.3	0.1	0.7	0.3	0.0	0.7	0.2	0.1
0.3	0.1	0.6	**0.3**	0.4	0.3	**0.7**	0.2	0.1
⋮			⋮			⋮		
0.7	0.0	0.3	0.8	0.1	0.1	0.6	0.2	0.2
0.0	0.9	0.1	**0.2**	0.7	0.1	**0.6**	0.3	0.1
0.5	0.1	0.4	**0.6**	0.2	0.2	**0.3**	0.3	0.4
0.6	0.1	0.1	0.7	0.3	0.0	0.7	0.2	0.1
0.7	0.2	0.1	**0.5**	0.2	0.2	**0.1**	0.8	0.1





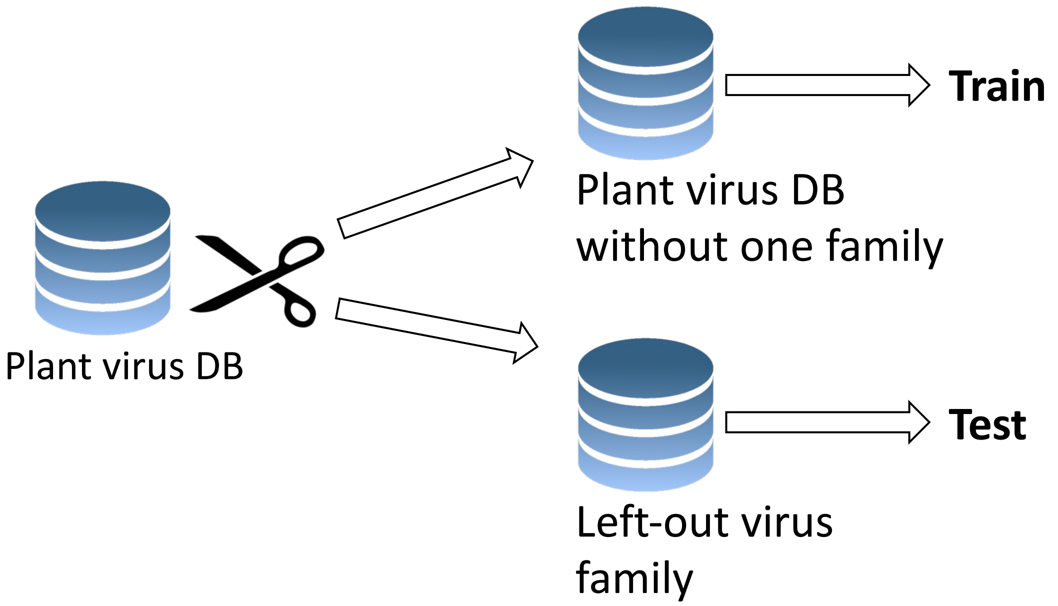


Potential of novel virus detection with VirHunter

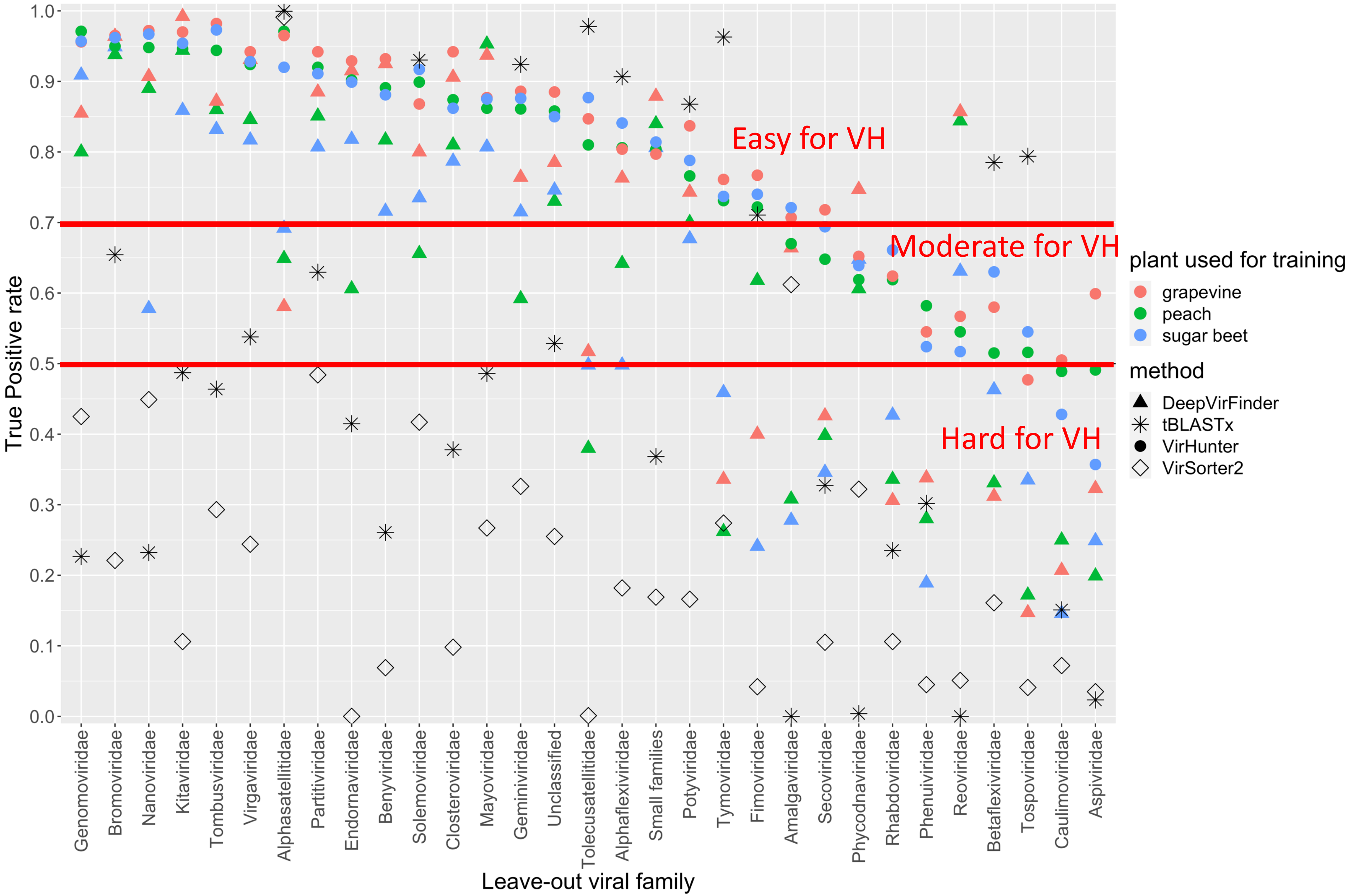
Evaluation of VirHunter: family leave-out datasets

- 31 artificial family leave-out datasets

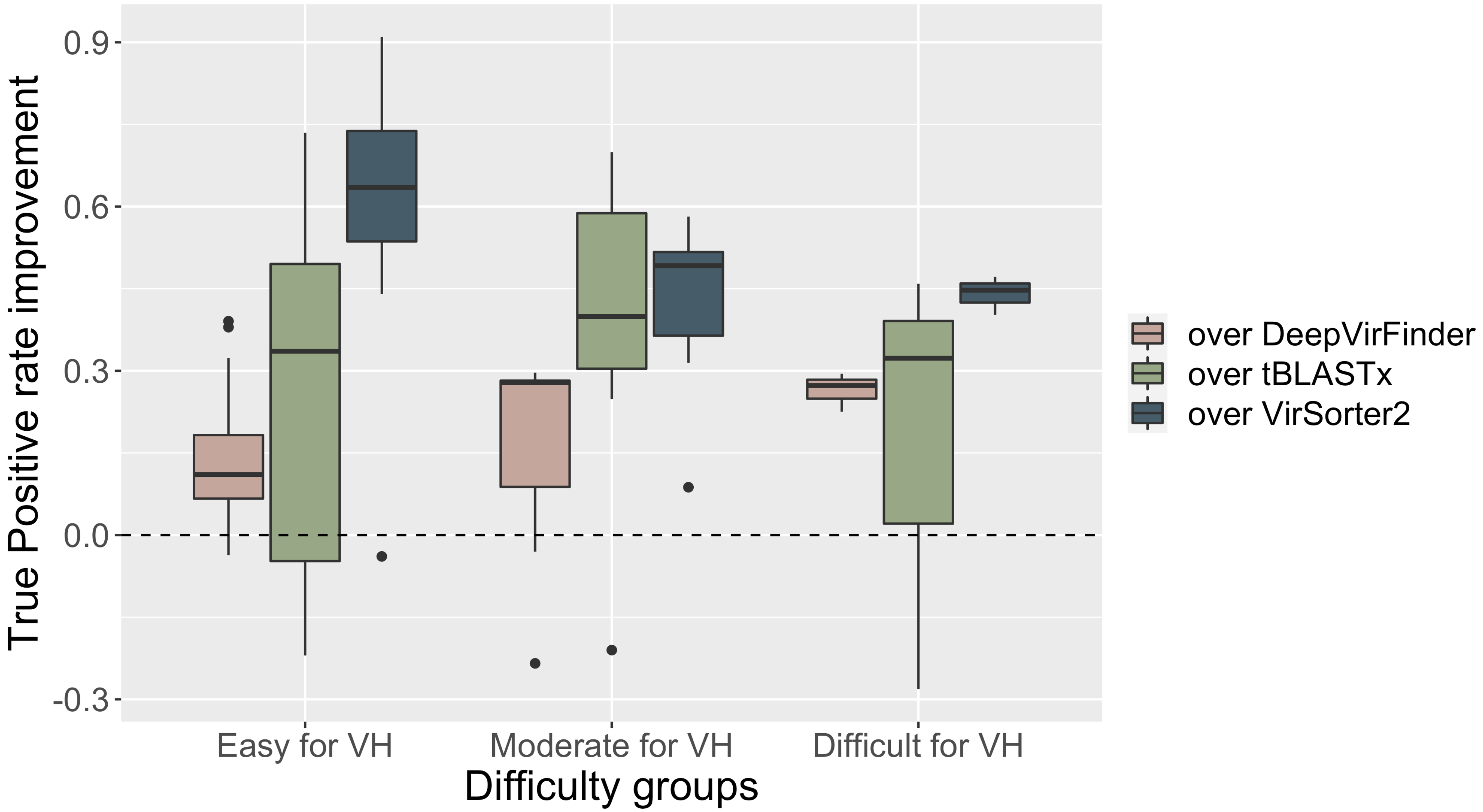
- Compared with:
  - DeepVirFinder
  - VirSorter2
  - tBLASTx



VirHunter improves over existing methods



VirHunter improves over existing methods



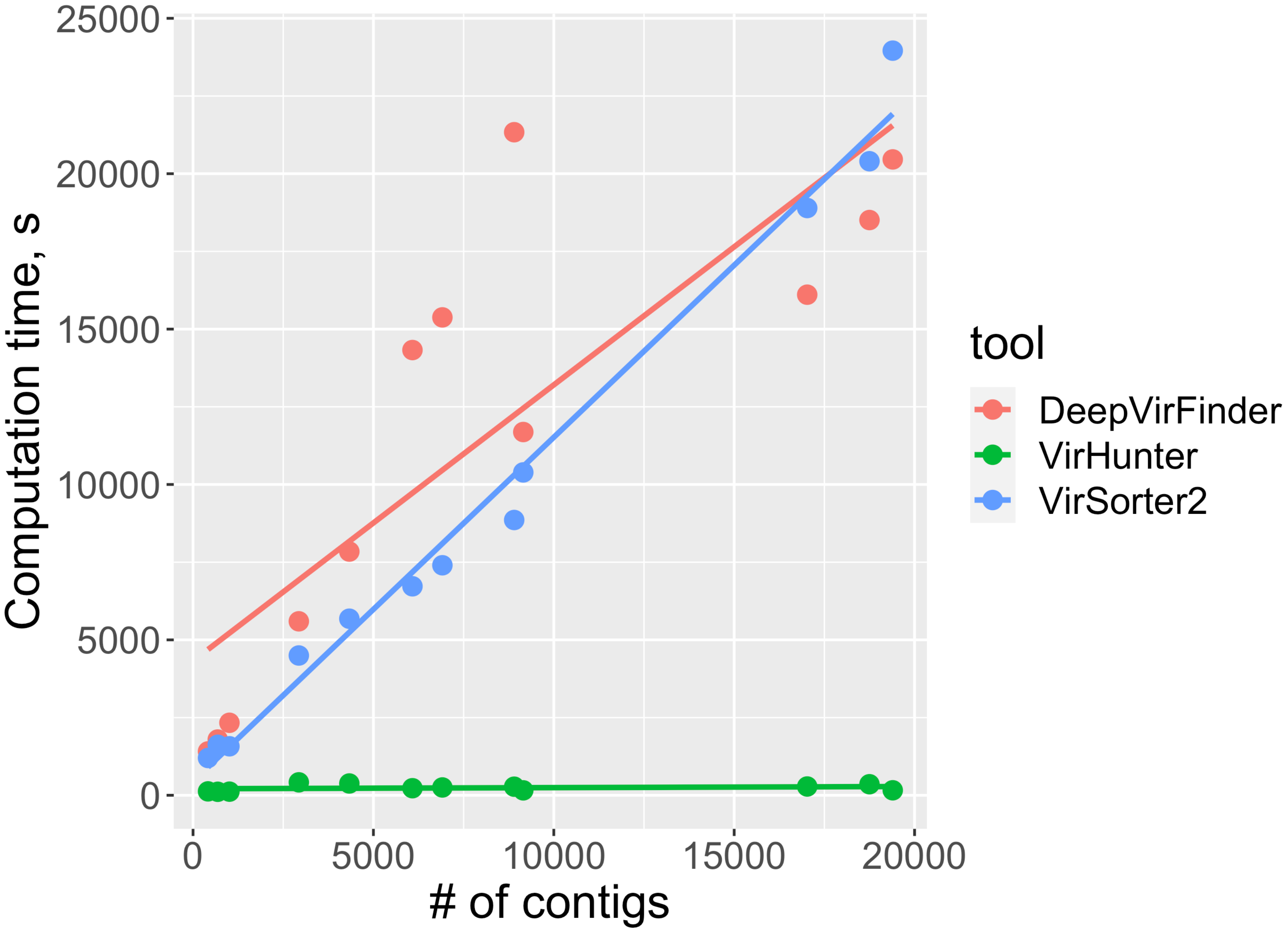
Evaluation of VirHunter: plant virome RNAseq samples

- 12 RNAseq datasets from peach, grapevine and sugar beet from INRAE Bordeaux-Aquitaine
- Viruses present in datasets were removed from Virus DB to simulate unknown virus detection

VirHunter detects most of annotated contigs

Dataset ID and plant origin		# contigs > 750	# contigs annotated as viral	VirHunter # detected (# annotated)	DeepVirFinder # detected (# annotated)	VirSorter2 # detected (# annotated)
P1	peach	1009	2	35 (2)	45 (2)	10 (1)
P2	peach	415	2	19 (2)	32 (2)	8 (1)
P3	peach	685	2	23 (2)	49 (2)	7 (1)
G1	grapevine	9154	10	153 (**10**)	133 (6)	52 (4)
G2	grapevine	17024	10	178 (**10**)	131 (9)	117 (6)
G3	grapevine	18750	20	208 (**18**)	137 (17)	142 (11)
G4	grapevine	4332	15	95 (**14**)	81 (11)	24 (4)
G5	grapevine	19395	25	262 (23)	302 (23)	144 (8)
G6	grapevine	2932	15	70 (**14**)	86 (13)	26 (12)
S1	sugar beet	6082	11	236 (10)	335 (**11**)	28 (6)
S2	sugar beet	8902	16	277 (16)	419 (16)	37 (7)
S3	sugar beet	6912	11	203 (11)	307 (11)	21 (4)

VirHunter is computationally efficient



Key points

- VirHunter detects well very divergent novel viruses (family leave-out datasets)
- It detects most of viral contigs in RNAseq datasets
- It is capable to deal with bacterial contamination
- It is fast

VirHunter: a deep learning-based method for detection of novel RNA viruses in plant sequencing data

Macha Nikolski[1, 2*], Grigorii Sukhorukov[2, 1*], Maryam Khalili[3], Olivier Gascuel[4], Thierry Candresse[3], Armelle Marais[3]

https://github.com/cbib/virhunter

frontiers in Bioinformatics

VirHunter

Practical aspects of VirHunter

VirHunter limitations

- Needs to be retrained for different plants
- Outputs confident prediction for contigs > 750 bp

VirHunter available models

- **Generalistic**
- Peach
- Apple
- Carrot
- Rice
- Sugar beet
- Grapevine
- Tomato

VirHunter example output

id	length	# viral fragments	# plant fragments	# bacterial fragments	decision	# viral / # total
contig_12	10871	21	0	0	virus	1.0
contig_72	5823	11	0	0	virus	1.0
contig_1725	5668	11	0	0	virus	1.0
contig_21	4230	8	0	0	virus	1.0
contig_1005	3121	6	0	0	virus	1.0
contig_468	3635	0	7	0	plant	0.0

To fasta file

How VirHunter fits into your pipelines?

- Quickly reduces number of contigs to study
- Detects novel viruses
- Provides support for other detection methods

How to install VirHunter?

- Installation with conda on MacOS and Linux
- Very soon to be available on Galaxy
- https://github.com/cbib/virhunter

ANACONDA®

Galaxy
EUROPE

Decontaminator

VirHunter’s friend – Decontaminator

- DL-based filtering step before VirHunter
- Filters out bacteriophages and fungi
- Reduces VirHunter’s overprediction

DECONTAMINATOR

VirHunter + Decontaminator

Dataset ID and plant origin		# contigs > 750	# contigs annotated as viral	VirHunter # detected (# annotated)	VirHunter + Decontaminator # detected (# annotated)
P1	peach	1009	2	35 (2)	19 (2)
P2	peach	415	2	19 (2)	7 (2)
P3	peach	685	2	23 (2)	11 (2)
G1	grapevine	9154	10	153 (10)	92 (10)
G2	grapevine	17024	1[illegible]	[illegible]3 (10)	132 (10)
G3	grapevine	18750	20	208 (18)	61 (18)
G4	grapevine	4332	15	95 (14)	79 (14)
G5	grapevine	19395	25	262 (23)	131 (23)
G6	grapevine	2932	15	70 (14)	48 (14)
S1	sugar beet	6082	11	236 (10)	116 (9)
S2	sugar beet	8902	16	277 (16)	143 (15)
S3	sugar beet	6912	11	203 (11)	127 (11)

↓ 47%