Chapter 16

Construction of a Comprehensive Database from the Existing Viral Sequences Available from the International Nucleotide Sequence Database Collaboration

Rodrigo García-López

Abstract

The progress in viromics research has led to the accumulation of a large number of sequences from different types of viruses obtained from different sources. Most databases are specific to different of species or types of viruses. However, raw sequences, as deposited in the reliable online collections, provide a valuable asset in the exploration of genomic and metagenomics datasets.

The International Nucleotide Sequence Database Collaboration (INSDC) is the largest coordinated effort for compiling, sharing, and maintaining the most comprehensive collections of nucleic acids deposited throughout the most important public databases. The compendium includes different types of data such as complete genomes, genes, expressed sequence tags, and data generated by whole genome shotgun analyses spanning all domains of life, as well as the most complete collection of viral sequences available online.

This chapter presents simplified computational methods for the automation of viral nucleotide sequence retrieval from online repositories of the INSDC databases, including all available sequences, except synthetic ones. The subsequent steps can be used for obtaining the taxonomy (including ranks: virus type, baltimore classification, order, family, subfamily, genus and species), and split the database into species subsets to dereplicate the sequences for other downstream applications. Only basic computational knowledge is required.

Key words Computational methods, Databases, Viruses, Genomes, Genes, Taxonomy

1 Introduction

Viromics is still a growing field that has been traditionally limited by reference sequence availability [1]. Although the collections of deposited viral sequences have increased exponentially in recent years (mainly due to the advent of metagenomics) there is still a large gap in the knowledge and exploration of whole viromic datasets, mostly due to the limited pool of reference viral sequences available [2]. Historically, studies have focused on specific species,

Andrés Moya and Vicente Pérez Brocal (eds.), *The Human Virome: Methods and Protocols*, Methods in Molecular Biology, vol. 1838, https://doi.org/10.1007/978-1-4939-8682-8_16,

most often phages affecting model organisms, human health or human-related commercial activities (crops and livestock), which makes up a large percentage of viral contents in public databases [3, 4]. Still, the ecological relevance of environmental viruses has been highlighted as viruses seem to play a relevant role in any biome [5]. This has promoted a deeper and thorough exploration of the viromic contents using whole genomic sequencing approaches, further emphasizing the need for larger and more inclusive viral databases.

Furthermore, viral taxonomy is commonly hindered by a classification system that is often host or pathology-related [6] as viruses share no universal marker and are rather a polyphyletic group [7, 8]. Their taxonomy is further complicated by the large prevalence of horizontal gene transfer since many act as mobile genomic elements that integrate into genomes, further increasing their genetic plasticity [9, 10]. As the naming authority, the International Committee on Taxonomy of Viruses (ICTV) has tackled the complex endeavor of revising classification but its rigid system has not coped well with the advent of high-throughput data from metagenomic studies [6–11].

Different tools have been created for improving viral classification but reference sequences continue to pose as one of the most important features for identification, whether for direct sequence search, or downstream construction of alignments and Hidden Markov Models [12, 13]. As a consequence, there are a number of nucleotide databases that are publicly available, some of which are manually curated with annotations, others just storing collections of genes or genomes [14, 15]. As vast as the number of applications for the sequences is, database selection is a critical process for viromic studies and, for all general purposes, having a comprehensive collection is most convenient.

The largest of its kind is the viral fraction of the nucleotide collection of the International Nucleotide Sequence Database Collaboration (INSDC), part of the synchronized effort of the European Bioinformatics Institute (EBI), DNA Data Bank of Japan (DDBJ), and the National Center for Biotechnology Information (NCBI) to make all their nucleotide sequences available to the general scientific community [16]. For over 30 years, the INSD has been the primary repository for nucleotide sequences that are submitted to any of its partners, coordinating the contents in mirrored images localized in dedicated servers [17], allowing for the preservation and access to data worldwide. Data is constantly exchanged and synchronized across the different servers for updates and fixes.

FTP servers are available for easier sequence access from all three partners, with different features residing in each server. The available datasets from the INSDC include the standard release of the main nucleotide sequence collection, which includes genes and

genomic sequences from all domains of life and viruses, as well as expressed sequence tags, genome survey sequences, high-throughput genome sequencing data, transcriptome shotgun assemblies, whole genome shotgun data, and patented items [17]. The identifiers, or accession numbers (shared among all partners in the INSDC) can be used to effectively trace the source organisms or viruses using cross-reference tables pointing to taxonomic identifiers or taxids. These are also available from the FTP servers, enabling the assembly of a complete viral taxonomy for any given sequence and can also be used to separate the sequences based on it, as it is described in the following methods.

2 Materials

2.1 Sequence Retrieval

1. PC with1. Linux 3 or higher (tested on Kubuntu 16.04 distribution).
   Alternatively, macOS or Windows 10 (with Windows subsystem for linux enabled).
2. Bash 4 or equivalent shell.
   As of Sept 2017, default in most current Linux distributions.
3. Over 10 free gigabytes (GB) disk space for storage for all methods (preferable: >20 GB).
   As of Sept 2017 a minimum of 1.2 GB is required for Subheading 2.1.
4. Stable internet connection (>1 Mbps is preferable).
   Current version requires downloading 631 MB of remotely stored data.
   IMPORTANT: All data should be downloaded in the same day to avoid inconsistencies due to updates.

- Optional: FileZilla 3.7 or higher.

Installation instructions and files at https://filezilla-project.org/.

2.2 Assembly of Taxonomic Ranks

1. All non-optional items in Subheading 2.1.
   As of Sept 2017, a minimum of 5.3 GB of additional disk space is required for Subheading 2.2.
   Current version requires downloading 4.8 GB of remotely stored data.
   IMPORTANT: All data should be downloaded on the same day to avoid inconsistencies due to updates.
2. Perl v5 of higher.
   As of Sept 2017, included in most current Linux distributions.

- Optional: File containing a list of Accession Numbers (one identifier per line)
- Optional: FileZilla 3.7 or higher.

Installation instructions and files at https://filezilla-project.org/

2.3 Database Split by Species and Dereplication

1. All non-optional items in Subheading 2.2.
   As of Sept 2017, a minimum of 4.2 GB of additional disk space is required for Subheading 2.3.
2. VSEARCH (distributed under GNU General Public License version 3; tested version: 2.4.4).

3 Methods

3.1 Accessing and Retrieving the Remote Files from the INSDC

1. Create a folder for allocating the database and access it (Important: hereafter, the prompt is marked with the initial **$** sign and is not part of the command; *see* **Note 1**):

```
$ wdir = ~/VirusDB_2017-09-18/;mkdir $wdir;cd $wdir
```

2. Automatic unsupervised download of all sequences (*see* **Note 2**):

```
$ for set in est gss htg std tsa; do for
type in vrl phg;do for stat in "release/
em_rel" "new/em_cum";do while [ ! -f "01_
INSD/${stat#*/}_${set}_$type.gz" ]; do echo
"Attempting download of ${stat#*/}_${set}_$type.
gz";wget -t 3 -T 2 --user = anonymous --no-
directories --directory-prefix = 01_INSD
ftp://ftp.ebi.ac.uk/pub/databases/fastafiles/
embl${stat}_${set}_${type}.gz 2 > temp.log; if
grep -q "ERROR 404" temp.log;then touch 01_
INSD/${stat#*/}_${set}_$type.gz;echo "${stat#*/}
$set $type does not exist (not an error)";else if
grep -q "Read error at byte" temp.log;then rm 01_
INSD/${stat#*/}_${set}_$type.gz; fi; fi; done; done;
done; done; rm temp.log;find ./01_INSD/ -type f
-empty -delete
```

3. Concatenate and fix headers (*see* **Note 3**):

```
$ for i in $(ls -tr 01_INSD);do y = ${i^^};zcat
01_INSD/$i|sed "s/^>\w*:\w*/>${y:3:11}/"|perl -lne
'next if $_ =~ /^$/;if ($_=~/>/){$line=$_;$line
=~ s/^>\K\w+:\w+ /$1/;($id,$rest)=split(/\
s/,$line,2);$rest =~ s/[\;\:]/ /g;$rest =~ s/[\[\
{]/\(/g; $rest =~ s/[\}\]]/\)/g;$rest =~ s/[^a-zA--
Z\d\s\(\)\.\-\_\/]//g;$rest =~ s/ +/ /g;$rest =~ s/
\.$//;$rest=substr($rest,0,200);print "$id $rest";}
else {print uc($_)}'|gzip >>INSD.gz;done
```

- [Optional: remove the source datasets (no longer required)]

```
rm. -r 01_INSD
```

3.2 Taxonomy

1. Create a folder for the taxonomy:
   $ *mkdir 02_Taxonomy*
2. Create a list of accession numbers from the database in a new folder (*see* **Note 4**):
   $ *zgrep ">" INSD.gz|cut -d " " -f 2|cut -d. -f 1|sort|uniq >02_Taxonomy/INSD.accs*
3. Get the official files for taxonomy from the NCBI taxonomy database (*see* **Note 5**):

   ```
   $ while [ ! -f "02_Taxonomy/taxdump.tar.gz" ];do
   echo "Attempting download of taxdump.tar.gz";wget
   -t 3 -T 2 --user=anonymous --no-directories
   --directory-prefix=02_Taxonomy ftp://ftp.ncbi.nlm.
   nih.gov/pub/taxonomy/taxdump.tar.gz 2>taxonomy_
   download.log; if grep -q "Read error at byte" tax-
   onomy_download.log;then rm 02_Taxonomy/taxdump.tar.
   gz;fi;done;rm taxonomy_download.log
   ```

4. Unpack only the usable files from the tar file (*see* **Note 6**)

   ```
   $ for i in nodes.dmp names.dmp; do tar -C 02_
   Taxonomy/ -xzf 02_Taxonomy/taxdump.tar.gz $i;done.
   ```

5. Unsupervised retrieval of the accession number to taxid cross-reference tables (acc2taxid) from the NCBI (*see* **Note 7**).

   ```
   $ for i in nucl_gb nucl_wgs nucl_gss nucl_est
   dead_nucl dead_wgs;do while [ ! -f "02_Taxonomy/$i.
   accession2taxid.gz" ];do echo "Attempting down-
   load of $i.accession2taxid.gz";wget -t 3 -T 2
   --user=anonymous --no-directories --directory-
   prefix=02_Taxonomy ftp://ftp.ncbi.nlm.nih.gov/pub/
   taxonomy/accession2taxid/$i.accession2taxid.gz
   2>taxonomy_download.log; if grep -q "Read error at
   byte" taxonomy_download.log;then rm 02_Taxonomy/$i.
   accession2taxid.gz;fi;done;done;rm taxonomy_down-
   load.log
   ```

6. Obtain the proprietary scripts for processing the taxonomy (*see* **Note 8**).

   ```
   $ wget https://github.com/rodrigogarlop/VirusDB_
   Scripts/raw/master/get_taxid_from_accnum_list.
   pl; wget https://github.com/rodrigogarlop/
   VirusDB_Scripts/raw/master/get_taxonomy_and_sptax-
   id_from_acc2taxid.pl; wget https://github.com/rod-
   rigogarlop/VirusDB_Scripts/raw/master/filter_pretax-
   onomy_general.pl
   ```

7. Use the acc2taxid cross-reference tables to obtain terminal node taxids (*see* **Note 9**).

```
$ perl get_taxid_from_accnum_list.pl 02_Taxonomy/
INSD.accs <(cat 02_Taxonomy/*.gz)
```

8. If any taxids are missing, retrieve additional taxids from the corresponding accession numbers using direct requests using the eutils tools from the NCBI servers (*see* **Note 10**). Also, create a list of known bad assignations.

```
$ if [ f 02_Taxonomy/INSD.accs.accs_with_no_tax-
id.txt ];then echo "Processing additional miss-
ing Accession numbers";while read i;do j=$(curl -s
"https://eutils.ncbi.nlm.nih.gov/entrez/eutils/
efetch.fcgi?db=nuccore&id=$i&retmode=text&retty
pe=native"|egrep -A 2 "taxon"|egrep "id \w"|cut
-d "d" -f 2|cut -d " " -f 2|head -n 1) ; printf
"%s\t%s\t%s\t%s\n" $i $i $j $j;done <02_Taxonomy/
INSD.accs.accs_with_no_taxid.txt >02_Taxonomy/last_
chance.txt; awk 'BEGIN {FS="\t"} $4!="" {print}'
02_Taxonomy/last_chance.txt >>INSD.accs.acc_tax-
id.txt;awk 'BEGIN {FS="\t"} $4=="" {print $1}'
02_Taxonomy/last_chance.txt >02_Taxonomy/failed.
txt;fi;printf "0\t10407\n1843481\t1299908\n1843482\
t1299909\n423445\t1891764\n1436891\t2038729\n" >02_
Taxonomy/fix_taxids.txt
```

9. Fix known bad assignations (may be edited accordingly; *see* **Note 11**).

```
$ perl -lane 'BEGIN {open(IN, "02_Taxonomy/
fix_taxids.txt")||die "Cannot open fix_taxids.
txt";while (<IN>){chomp $_;@line = split(/\
t/,$_);$fix{$line[0]}=$line[1] if $line[1] ne
""}}if(exists $fix{$F[2]}) {print "$F[0]\t$F[1]\
t$fix{$F[2]}\t$F[3]"}else{print $_};' 02_Taxonomy/
INSD.accs.acc_taxid.txt >02_Taxonomy/acc2taxid_All_
VirusDB.fix
```

10. Assemble the 22-category taxonomy using terminal-node taxids and create a table with the species-level taxids (*see* **Note 12**).

```
$ rm -f 02_Taxonomy/acc2taxid_All_VirusDB.fix.
bad;perl get_taxonomy_and_sptaxid_from_acc2taxid.
pl 02_Taxonomy/acc2taxid_All_VirusDB.fix;mv 02_
Taxonomy/acc2taxid_All_VirusDB.fix.sptaxid INSD.
sptaxid
```

11. [Important Optional Step: fix outdated taxids] The creation of file *acc2taxid_All_VirusDB.fix.bad* means some node information must be updated in the cross-reference tables. This can be fixed semi-automatically with the following commands followed by repeating commands from **step 9** onward. If this is not corrected, the taxonomy for the corresponding records will be void of taxonomic information and will thus be presented as unidentified sequences (*see* **Note 13**):

```
$ for taxid in $(cat 02_Taxonomy/acc2taxid_All_
VirusDB.fix.bad|cut -d " " -f 6);do i=$(grep -m 1 -w
"$taxid" 02_Taxonomy/acc2taxid_All_VirusDB.fix|cut
-f 1); j=$(curl -s "https://eutils.ncbi.nlm.nih.
gov/entrez/eutils/efetch.fcgi?db=nuccore&id=$i&ret
mode=text&rettype=native"|egrep -A 2 "taxon"|egrep
"id \w"|cut -d "d" -f 2|cut -d " " -f 2|head -n
1);printf "$taxid\t$j\n";done >>02_Taxonomy/fix_tax-
ids.txt
```

12. Retrieve viral ranks and append Baltimore classifications and Virus Type for a 7-categories taxonomy (*see* **Note 14**).

```
$ perl 02_Taxonomy/filter_pretaxonomy_general.pl
02_Taxonomy/acc2taxid_All_VirusDB.fix.pretaxonomy
>INSD.taxonomy.
```

13. [Optional: remove the taxonomy tables and intermediate steps (no longer required)].

```
$ rm -r 02_Taxonomy
```

3.3 Create Separate Datasets for Each Species and Reduce Redundancy

1. Obtain the proprietary scripts for separating the database (*see* **Note 15**).

```
$ wget https://github.com/rodrigogarlop/VirusDB_
Scripts/raw/master/split_fastaDB_by_taxid.pl
```

2. Using the compressed sequence file (from **step 3** in Subheading 3.1) and the species taxonomy file (from **step 11** in Subheading 3.2), create separate sequence files organized by species taxid (*see* **Note 16**).

```
$ perl split_fastaDB_by_taxid.pl INSD.sptaxid INSD.
gz
```

3. Define the pathway where VSEARCH is installed (IMPORTANT: change accordingly; *see* **Note 17**):

```
$ vsearch=/path/to/vsearch
```

4. Create a report containing the species taxid, the number of sequences, the total bp, and the taxonomy for each species (*see* **Note 18**).

```
$ printf "#Sptaxid\tSeqs\tSize\tbp\tType\tBalti-
more\tOrder\tFamily\tSubfamily\tGenus\tSpecies\n"
>Raw_composition.txt ;for i in $(ls -Srh 03_
BuildDB/|cut -d. -f 1);do seqs=$(grep -c "^>"
03_BuildDB/$i.fasta|tr -d '\012\015');size=$(ls
-lh 03_BuildDB/$i.fasta|sed 's/\s/ /g' |sed 's/
/\t/g'|cut -f 5);name=$(grep -wm 1 "^$i" INSD.
taxonomy);bp=$(grep -v "^>" 03_BuildDB/$i.
fasta|wc|awk '{print $3-$1}');printf "$i\t$seqs\
t$size\t$bp\t$name\n";done >Raw_composition.txt
```

5. Reduce redundancy by dereplicating the datasets and summarize results (*see* **Note 19**):

```
$ rm -f derep_INSD.fasta;printf "#Sptaxid\tSeqs\
tSize\tbp\tType\tBaltimore\tOrder\tFamily\tSubfam-
ily\tGenus\tSpecies\n" >Derep_composition.txt;for
i in $(ls 03_BuildDB/|cut -d. -f 1);do seqs=$(grep
-c "^>" 03_BuildDB/$i.fasta);if [ $seqs -eq 1
];then cat 03_BuildDB/$i.fasta >03_BuildDB/$i.
derep.fasta;else $vsearch --derep_fulllength 03_
BuildDB/$i.fasta --fasta_width 80 --output 03_
BuildDB/$i.derep.fasta --maxseqlength 1000000000000
--notrunclabels;fi;cat 03_BuildDB/$i.derep.fas-
ta >>INSD_derep.fasta;seqs=$(grep -c "^>" 03_
BuildDB/$i.derep.fasta|tr -d '\012\015');size=$(ls
-lh 03_BuildDB/$i.derep.fasta|sed 's/\s/ /g'
|sed 's/ /\t/g'|cut -f 5);name=$(grep -wm 1
"^$i" INSD.taxonomy|cut -f 2-);bp=$(grep -v
"^>" 03_BuildDB/$i.derep.fasta|wc|awk '{print
$3-$1}');printf "$i\t$seqs\t$size\t$bp\t$name\n"
>>Derep_composition.txt;rm 03_BuildDB/$i.derep.
fasta;done
```

4 Notes

1. All computational methods presented here should work by inputting them directly in the command line. The prompt (initial $), should not be copied, it just marks where the command begins.
2. In this method, the sequences from the INSDC are downloaded from the European Bioinformatics Institute as it is readily accessible, it is updated weekly and uses an adequate header format (accession with version and description; since September, 2016, the National Centre for Biotechnology Information uses the same format). The last release is found in ftp.ebi.ac.ukftp.ebi.ac.uk/pub/databases/fastafiles/emblrelease/ and contains the cumulative collection whereas the updates are located at ftp.ebi.ac.ukftp.ebi.ac.uk/pub/databases/fastafiles/emblnew/

   The INSDC virus and bacteriophages files are identified by the vrl and phg suffixes respectively and are contained in the following sets:

   EST: Expressed Sequence Tag

   GSS: Genome Survey Sequence

   HTG: High-Throughput Genome sequencing

   STD: Standard

   TSA: Transcriptome Shotgun Assembly

   PAT: Patent

Patented items are not included in these procedures as the exact sequences are not expected to occur naturally or non-patented variants are preferred.

Please note that not all sets may have updates (a possibility that is considered by the download command). The command will continuously resubmit download requests if the connection is not established correctly unless no file exists in the first place.

As an alternative solution, files may be downloaded using a ftp client such as FileZilla, which is recommended for unstable connections. Please make sure they are located in a folder named *01_INSD* and that the subsequent commands are run from the base folder (*../ 01_INSD;* not from within the *01_INSD* folder).

3. As sequence descriptors in the headers have an unrestricted format, several sequences bear rare characters that must be removed to avoid parsing problems downstream, most notably whenever there is more than one ">"sign (which marks the start of the header/seq duplex). All rare characters are changed to spaces and then collapsed into single spaces. Brackets and the sort are changed to parentheses. Headers longer than 200 characters are truncated as they are also problematic. All sequences are changed to uppercase letters so beware when using masked sequences. Additionally, the name of the subset is added for maximizing the traceability.
4. The assembly of a ranked taxonomy starts with a sequence bearing an accession number as this identifier is linked to a taxonomic identifier (taxid). A single-column file containing these numbers is generated. The version in the accession numbers is ignored to avoid conflicts with outdated items.
5. The taxdump.tar.gz file contains several tools for working out the taxonomy assigned to taxid. It can be downloaded from ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/ (optionally with FileZilla).
6. The nodes.dmp is a file containing the links between different taxonomic nodes (from specific branching nodes to the more general root, always in a one-to-one relation) whereas the names.dmp contains the scientific names assigned to each taxid.
7. Cross-reference tables contain sets of accession numbers and their annotated taxids as well as NCBI's GIs. Although it is possible to download the whole cross-reference tables, it is unadvisable under unstable network conditions. The usage of FileZilla in this step is thus encouraged. To continue, please make sure the files are placed in a folder named *02_Taxonomy* and that the subsequent commands are run from the base

folder (*../*02_Taxonomy*;* not from within the *02_Taxonomy* folder).

Protein accession number to taxids cross-reference tables also exist in remote servers.

8. The perl scripts are freely available from github and were created by author Rodrigo García-López and distributed under a GNU 3 license.
9. The script was optimized for low memory consumption and it normally takes ~20–25 min to finish. It stores the loads all accession numbers found in the current downloaded sets and gets those in the taxonomy cross-reference tables where there is a coincidence.

   There is no need to unzip any of the tables.
10. Some acc_nums may have no traceable taxid in the tables but they can be obtained with a remote request to the entrez service via html accession with eutils. This command parses the corresponding remote xml files, which may be slow in some cases but provides updated versions of the accession numbers.
11. Some errors have been reported to exist in the official tables (as downloaded from the NCBI). These must be addressed prior to constructing the taxonomy to avoid bad assignations. This command uses the auxiliary *fix_taxids.txt* created in the previous step, which may be changed according to the missing taxids in the current database (as some may get corrected over time). These types of errors are uncommon but exist mostly permanently in the remote servers.

    IMPORTANT: To avoid some of these inconsistencies, it is advisable to download the sequence data on the same day as the cross-reference tables, since terminal nodes may change when unknown sequences get identified or described, rendering them untraceable.
12. Standardized taxonomies are made of 22 ranked categories, most of which are normally empty for viruses, plus several unranked classifications. Each category is defined by a taxonomic identifier (taxid) which is related to a taxonomic rank (or "no rank"), a scientific name and the link to the next node in the tree (one with higher hierarchy). To assemble the complete taxonomy, the script loads the complete list of links between taxids, the corresponding names and the ranks, and uses the terminal node taxid that is related to an specific accession number (retrieved in previous steps) as the starting point. Each taxid is traced back to its highest level (root of the tree) one node at a time and those bearing ranked categories are organized, effectively constructing a 22 categories taxonomy plus a string containing all unraked categories.

Not all initial taxids point toward the species level but to terminal nodes in the taxonomy, which may be in even lower levels (e.g., serovar/serotype) or just unidentified sequences from known species. The script thus creates an additional list containing the accession number, species-level taxid (sptaxid), and terminal-node taxid (these last two are the same in most cases). The sptaxid is the one used for splitting the database downstream in Subheading 3.3.

This methodology can also be used to process any eukaryotic or prokaryotic sequence, as long as accession numbers and taxids are provided.

13. When taxids get updated (e.g., when unidentified sequences are classified into existing categories) then the terminal taxids change. To avoid this, the nodes.dmp and names.dmp files should be downloaded on the same date as tables from the acc2taxid cross-reference tables (and preferably, on the same date as the sequence data as well). In rare cases, some of them may be outdated. In order to address this, a remote request to the entrez can be carried out with the accession number to retrieve updated taxids. This constructs the input files to append them to the *fix_taxids.txt* file and re-run all commands from **step 9**. Repeat until no *acc2taxid_All_VirusDB.fix.bad* is created in **step 11**, Subheading 3.2.
14. The pretaxtonomy created in **step 10**, Subheading 3.2 contains the 22 ranked categories plus a string of unraked classifications. The script filters them and adds the Baltimore Classification and virus type classification (from unranked categories), additionally dealing with empty categories by including the nearest level that has relevant information for maximum traceability. The output from this step creates the usable taxonomy, containing the related taxid (the one that is found in the sequence headers) and 7 taxonomic categories: Virus type, Baltimore, order, family, subfamily, genus, and species.
15. The perl script is freely available from github and was created by author Rodrigo García-López and distributed under a GNU 3 license.
16. This script is used to organize a large fasta file (.gz) by species based on the accession numbers that are present as the second item in the previously formatted sequence headers (space-separated; e.g., >REL_GSS_VRL AF079492.1 AF079492 Rhesus cytomegalovirus genomic DNA sequence). To do so, it requires the sptaxid crossreference table created in earlier steps which contains three tab-separated columns with accession numbers, species taxids (sptaxids), and terminal-node taxids (commonly matching the sptaxids).

Each newly created fasta file is named after the species taxids contained within (18,621 files in total as of September, 2017).

To do a terminal-node separation instead pass the following alternate command: $ perl split_fastaDB_by_taxid.pl <(cut -f 1,2 INSD.sptaxid) INSD.gz

17. VSEARCH [18] is an open-source alternative to USEARCH [19] suite by Robert C. Edgar that can be used to dereplicate and cluster sequence databases. This step defines the complete pathway of the program's executable file, which depends on the program's installation. Alternatively, USEARCH may be used as the commands are the same (tested version: 8.0).
18. The report table contains the following items for each file (species): taxid, total sequences, used disk space (human readable), total bp, virus type, Baltimore classification, order, family, subfamily, genus, species.
19. By default, VSEARCH discards sequences longer than 50,000 nucleotides. This is overridden by increasing lower limit value with the --maxseqlength option. This approach will swiftly clear off exact identical matches but VSEARCH can also be used to cluster together different sequences with similar identities using the --cluster_fast (memory intensive).

References

1. Beerenwinkel N, Günthard HF, Roth V et al (2012) Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. Front Microbiol 3:1–16
2. Pérez-Brocal V, García-López R, Vázquez-Castellanos JF et al (2013) Study of the viral and microbial communities associated with Crohn's disease: a metagenomic approach. Clin Transl Gastroenterol 4:e36
3. Morgan GJ (2016) What is a virus species? Radical pluralism in viral taxonomy. Stud Hist Phil Biol Biomed Sci 59:64–70
4. Calisher CH (2016) The taxonomy of viruses should include viruses. Arch Virol 161:1419–1422
5. Cobián Güemes AG, Youle M, Cantú VA et al (2016) Viruses as winners in the game of life. Annu Rev Virol 3(1):197–214
6. Van Regenmortel MH, Ackermann HW, Calisher CH et al (2013) Virus species polemics: 14 senior virologists oppose a proposed change to the ICTV definition of virus species. Arch Virol 158:1115–1119
7. Edwards R, Rohwer F (2005) Viral metagenomics. Nat Rev Microbiol 3:504–510
8. Proux C, van Sinderen D, Suarez J et al (2002) The dilemma of phage taxonomy illustrated by comparative genomics of Sfi21-like Siphoviridae in lactic acid bacteria. J Bacteriol 184:6026–6036
9. Gibbs AJ (2013) Viral taxonomy needs a spring clean. Its exploration era is over. Virol J 10:254
10. Moreira D, López-García P (2009) Ten reasons to exclude viruses from the tree of life. Nat Rev Microbiol 7:306–311
11. International Committee on Taxonomy of Viruses, ICTV Species List 2016 v 1.2, (https://talk.ictvonline.org/)
12. Roux S, Emerson JB, Eloe-Fadrosh EA et al (2017) Benchmarking viromics: an in silico evaluation of metagenome-enabled estimates of viral community composition and diversity. PeerJ 5:e3817
13. Watkins SC, Putonti C (2017) The use of informativity in the development of robust viromics-based examinations. PeerJ 5:e3281
14. Pickett BE, Sadat EL, Zhang Y et al (2012) ViPR: an open bioinformatics database and analysis resource for virology research. Nucleic Acids Res 40:593–598

15. Liechti R, Gleizes A, Kuznetsov D et al (2010) OpenFluDB, a database for human and animal influenza virus. Database (Oxford) 2010:baq004
16. Cochrane G, Karsch-Mizrachi I, Takagi T (2016) The international nucleotide sequence database collaboration. Nucleic Acids Res 44(D1):D48–D50
17. Nakamura Y, Cochrane G, Karsch-Mizrachi I (2013) The international nucleotide sequence database collaboration. Nucleic Acids Res 41(Database issue):D21–D24
18. Rognes T, Flouri T, Nichols B et al (2016) VSEARCH: a versatile open source tool for metagenomics. PeerJ 4:e2584
19. Edgar RC (2010) Search and clustering orders of magnitude faster than BLAST. Bioinformatics 26(19):2460–2461