Sequence analysis

VecScreen_plus_taxonomy: imposing a tax(onomy) increase on vector contamination screening

Alejandro A. Schäffer*, Eric P. Nawrocki, Yoon Choi, Paul A. Kitts, Ilene Karsch-Mizrachi and Richard McVeigh

Department of Health and Human Services, National Center for Biotechnology Information, National Institutes of Health, Bethesda, MD 20894, USA

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on August 9, 2017; revised on September 18, 2017; editorial decision on October 17, 2017; accepted on October 21, 2017

Abstract

Motivation: Nucleic acid sequences in public databases should not contain vector contamination, but many sequences in GenBank do (or did) contain vectors. The National Center for Biotechnology Information uses the program VecScreen to screen submitted sequences for contamination. Additional tools are needed to distinguish true-positive (contamination) from false-positive (not contamination) VecScreen matches.

Results: A principal reason for false-positive VecScreen matches is that the sequence and the matching vector subsequence originate from closely related or identical organisms (for example, both originate in *Escherichia coli*). We collected information on the taxonomy of sources of vector segments in the UniVec database used by VecScreen. We used that information in two overlapping software pipelines for retrospective analysis of contamination in GenBank and for prospective analysis of contamination in new sequence submissions. Using the retrospective pipeline, we identified and corrected over 8000 contaminated sequences in the nonredundant nucleotide database. The prospective analysis pipeline has been in production use since April 2017 to evaluate some new GenBank submissions.

Availability and implementation: Data on the sources of UniVec entries were included in release 10.0 (<ftp://ftp.ncbi.nih.gov/pub/UniVec/>). The main software is freely available at https://github.com/aaschaffer/vecscreen_plus_taxonomy.

Contact: aschaffe@helix.nih.gov

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

To aid in cloning and sequencing a biological nucleotide sequence, other sequences (cloning vectors, adaptors, linkers, etc.) may be ligated to the biological sequence. For simplicity, we usually refer to all those other sequences henceforth as ‘vectors’. Vectors may come from a combination of biological and artificial (i.e. synthesized in a laboratory) sources. The sequencing process produces a sequence that can include both the targeted biological sequence and one or more vector segments; usually, the non-targeted sequences are on

one end (called ‘terminal’), but the vector segments can also be in the middle (‘internal’) (Kim *et al.*, 2016). When the sequence is submitted to a public database, such as GenBank at the National Center for Biotechnology Information (NCBI), it is important for the submitter to trim off the vector segments and submit only the targeted sequence. More than two decades of experience with submissions to GenBank have shown that submitters occasionally fail to trim vectors completely. Therefore, NCBI screens all nucleotide submissions for *vector contamination*.

The standard procedures for screening nucleotide submissions at NCBI use a variant of the program *blastn* (Altschul *et al.*, 1997), called *VecScreen* (<https://www.ncbi.nlm.nih.gov/tools/vecscreen/>), which compares the submitted nucleotide sequence(s) as a query to a database of vector segments called *UniVec*. *VecScreen* uses a *blastn*-style scoring system that strongly penalizes differences other than the occasional mismatch or short gap expected from sequencing errors: match = +1, mismatch = -5, gap open cost = 3, gap extension cost = 3; by convention, the gap open and gap extension costs are shown as positive numbers, although they get subtracted when scoring an alignment. The *VecScreen* command-line application does not currently offer the option to change the scoring system and we did not explore changing the scoring system as part of this study. Expertise of biologists has been used to determine which submission-vector matches found by *VecScreen* represent true vector contamination. Here, we present a package, *VecScreen_plus_taxonomy*, with the intent of encoding in data and software much of the knowledge that biologists use to classify *VecScreen* matches, to make the classification into true and false matches more automatic and more deterministic.

In *VecScreen*, a match to a sequence is defined as *terminal* if it includes a nucleotide within 25 positions of either end of the sequence and *internal*, otherwise (<https://www.ncbi.nlm.nih.gov/tools/vecscreen/about/>). *VecScreen* also classifies matches by strength: a match is *strong* if it is terminal with a *blastn* raw score of at least 24 or internal with a raw score of at least 30. A match is *moderate* if it is terminal with a score in the interval 19–23 or internal with a score of 25–29. A match is *weak* if it is terminal with a score of 16–18 or internal with a score of 23–24. While terminal, weak matches (scoring 16–18) can often be false positives due to the *random* occurrence of identical oligomers in the sequence and a matching vector, higher-scoring *VecScreen strong* or *moderate* matches are rarely random. However, these higher-scoring matches may be false positives (non-vector) for reasons that can be detected systematically. *VecScreen_plus_taxonomy* aims to identify and discriminate these false positives from true-positive vector matches.

Examination of over 10 000 plausible *VecScreen* matches revealed that virtually all false-positive matches are false for one of three reasons:

1. the matching segments of the biological sequence and vector sequence share some taxonomic origin. For example, segments from the *UniVec* entry U03992.1: 408-1934 originate from *Pseudomonas aeruginosa* and segments from the *UniVec* entry U03498.1: 4889-5243 originate from *Saccharomyces cerevisiae*. Therefore, if a submitted sequence is from the genera *Pseudomonas* or *Saccharomyces* and matches the corresponding vector segments, those matches are judged not to represent true vector contamination and the matches are false positives.
2. The biological sequence is of bacterial origin and the matching vector segment is derived from a genomic region that confers antimicrobial resistance (AMR). AMR regions are commonly used in vectors because they allow selection of clones using common antibiotics. For biological and epidemiological reasons, AMR segments can jump in nature across a broad taxonomic spectrum. Hence, AMR segments are sometimes found in bacterial species divergent from the original genus. Therefore, the AMR type of false positive is taxonomically different from the genus-specific false-positive category 1.
3. The match is terminal, weak and random. The intent of *VecScreen_plus_taxonomy* is mostly to distinguish the first two categories of false positives from true positives. When the

random type arises, we record that instance, so it will be recognized as a false positive in the future. However, we did not systematically attempt to find all short vector segments that lead to weak, random matches.

The identification of the first type of false positive listed above takes advantage of taxonomic information of both the vector sequences and the sequences being screened for contamination. This idea that taxonomy is relevant to distinguishing true vector contamination from false matches was suggested indirectly in several previous attempts to find vector contamination in public sequence databases (Binns, 1993; Lamperti *et al.*, 1992; Lopez *et al.*, 1992; Miller *et al.*, 1999; Savakis and Doelz, 1993; Seluja *et al.*, 1999). For example, Seluja *et al.* (1999) did a retrospective screening for contaminated sequences, but they explicitly avoided screening *bacterial* sequences because of the high false-positive rate they would have had with the many vector segments that originate in bacteria. Viral sequences also have a high false-positive rate because many vectors have segments originating from viruses. A particularly complicated category includes vectors derived from retroviruses (Coffin *et al.*, 1997; Völter *et al.*, 1998), which can integrate into the genomes of eukaryotes. Thus, retrovirus-derived vectors have two sources: the virus and the host, such that matches to either are deemed false positives. Similarly, integrating bacteriophages have two sources: the viral genus of the phage and the bacterial genus of the host.

The idea that false-positive contamination could be recognized with the aid of pre-constructed databases of sequence sources is not novel. The software package *DeconSeq* for microbial metagenomics (Schmieder *et al.*, 2011) is based on this concept. In that context, ‘contamination’ refers to DNA from the human collectors of samples mixing with the microbial specimens, not vector contamination. White *et al.* (1993) presented a database-free statistical method to detect such cross-species contamination of samples.

2 Materials and methods

Using targeted *blastn* (Altschul *et al.*, 1997; Camacho *et al.*, 2009) analysis between vector segments and whole genomes of likely vector sources, we compiled a list of intervals of vector sequences in *UniVec* with known sources at genus level. For bacteriophages and retroviruses, the second host source was added by biological expert knowledge. We augmented this list with expert knowledge of vector sources for which a whole genome sequence does not exist, such as the luciferase gene of *Renilla*. While doing retrospective analysis of possible contamination of GenBank, we identified vector segments that are conserved at taxonomy levels higher than genus and recorded these higher levels, using biologists’ expert knowledge. For example, the five-column entries:

U13843.1:1-1126	405	1064	10088	Mus
U13843.1:1-1126	405	1064	9989	Rodentia

indicate that there is a vector segment occupying positions 1 through 1126 of sequence U13843.1, that the interval from 405 through 1064 originates in the genus *Mus*, whose NCBI taxid is 10088, and that this sequence interval is conserved in the order *Rodentia*, whose NCBI taxid is 9989. Therefore, using *VecScreen_plus_taxonomy*, any *VecScreen* match to this vector interval in which the sequence *S* comes from any genus (e.g. *Cavia*, *Hydrochoerus*, *Rattus*) within the order *Rodentia* would be classified as *FALSE_BIOLOGICAL*, which is the output annotation for matches that are false positives in category 1 listed in the Introduction section.

We identified whole *UniVec* entries (typically, adaptors and linkers) and *UniVec* entry segments that are artificial (no biological

source; the NCBI taxid is 81077 in the fourth column of the format shown above) from lists in (Coker and Davies, 2004; Falgueras *et al.*, 2010) as well as in artificial adaptor sequences listed at various resources (summarized at <http://onetippperday.sterding.com/2012/08/three-ways-to-trim-adaptorprimer.html>). Any VecScreen match to an artificial vector segment must be a true positive or a weak, random match. Such matches are classified as TRUE_ARTIFICIAL, unless we recorded the matching interval as a cause of weak, random biological matches.

Using blastn to compare UniVec (as queries) to an in-house database of AMR segments, we similarly compiled a list of vector segments at least 16 nucleotides long that match known AMR regions. Matches to these regions are false positive in category 2 in the Introduction section and are classified as FALSE_AMR.

We corrected problematic sequences by one of the first five out of six processes that are not mutually exclusive. 1) the vector segment can be trimmed out of the submitted sequence; 2) the sequence can be moved to the synthetic division of GenBank or otherwise annotated as having multiple biological sources; 3) other aspects of the sequence annotation, such as the definition line, can be corrected; 4) the sequence can be unverified (meaning that it remains in GenBank, but is considered suspect); 5) the sequence can be suppressed or 6) NCBI personnel can contact the submitter for an explanation. Contacting the submitter is the most preferred course of action for prospective analysis, but is the least preferred for retrospective analysis because the contact information from old submissions is often outdated. Therefore, we did not contact the submitter for any of the retrospective corrections done in our study.

In the first sentence of the previous paragraph, we used the adjective ‘problematic’ rather than ‘contaminated’ because some matches that get classified as TRUE_BIOLOGICAL have a problem that is not accidental contamination. The canonical examples are sequences in which a retroviral vector, which has some segment in UniVec, was inserted deliberately by an experimenter into a eukaryotic sequence. By GenBank policy, these laboratory-generated composite sequences belong in the synthetic division. We detected them as contaminated partly because our retrospective analysis considered only sequences not already in the synthetic division of GenBank (see [Supplementary File S1](#)).

For users interested in doing prospective analysis of their own sequence sets of reasonable size (e.g. fewer than 100 000 sequences), the main software and one version of the UniVec source data information are freely available on GitHub at https://github.com/aaschaffer/vecscreen_plus_taxonomy. [Figure 1](#) shows the logic of the overlapping pipelines that use VecScreen_plus_taxonomy. The logic in the arrow labeled ‘1’ of [Figure 1](#) is encoded in the package `generate_vecscreen_candidates`, which we distribute separately (https://github.com/aaschaffer/generate_vecscreen_candidates) because it is not needed by most users. This package does pre-processing to generate candidate sequences that are more likely to have VecScreen matches. The pre-processing is not essential, but is more efficient than running VecScreen on all sequences, if the database is large. The potentially contaminated sequences are then used as the query for VecScreen. The box labeled Query sequence(s) is where VecScreen_plus_taxonomy usage starts. All incoming submissions (prospective) to GenBank are used as Query sequences for VecScreen. If multiple VecScreen matches to a query are returned, each match is evaluated individually, which is why all boxes in [Figure 1](#) after VecScreen refer to matches in the query rather than to the query sequence. In practice, very few sequences of under 10 000 nucleotides have both a true-positive VecScreen match in one interval and a false-positive match in a different interval, but such mixed outcomes become more likely for longer sequences. All matches

must be negative to mark a query sequence as free of vector contamination.

To enable VecScreen_plus_taxonomy to be used both within the existing NCBI software architecture and as a standalone pipeline by any user, the pipeline is logically divided into two parts called `from_vecscreen_to_summary.pl` and `compare_vector_matches_wtaxa.pl`. The work in the boxes called Query sequence(s), VecScreen and Collect query source info is done in `from_vecscreen_to_summary.pl`, which we think of as the second part in retrospective analysis. The third part of retrospective analysis is the steps after Collect query source info; those later steps are implemented in `compare_vector_matches_wtaxa.pl`. Therefore, the two boxes that start the logic of `compare_vector_matches_wtaxa.pl` are labeled with a ‘3’ in [Figure 1](#).

Full instructions on how to invoke VecScreen_plus_taxonomy and examples can be found in the README file distributed with the software. The source information is also available in a more compact format with UniVec (<ftp://ftp.ncbi.nih.gov/pub/UniVec/>) starting with release 10.0

Most of the analysis and testing described above was done with UniVec version 9.0, which contains 5456 vector segments; where possible, we updated vector source files to UniVec 10.0, which contains 6093 vector segments. A single vector may have many disjoint segments in UniVec with different sources. For example, JN874480.1 (pHUGE-Red) has 31 segments, and those segments come from at least 10 distinct genera, representing viruses, bacteria, and plants.

There are 388 new sequences represented in UniVec 10.0; some of these are represented by multiple entries that are non-overlapping intervals within the sequence. There are also nine sequences that are newer versions (in the sense of GenBank sequence version) of sequences that were already represented in UniVec 9.0. Because new UniVec versions are developed by sequential addition of novel segments, there are some changes to the spans for vector segments that are represented in both UniVec 9.0 and 10.0. The development of VecScreen_plus_taxonomy adds extra files of source data for the sequence segments in UniVec 10.0, but no new data fields were added in the principal UniVec 10.0, FASTA-formatted file of sequence segments. Because UniVec 10.0 has fully superseded UniVec 9.0 in practice, we distribute only the source files for UniVec 10.0 with VecScreen_plus_taxonomy.

3 Results and discussion

We implemented in Perl a three-part pipeline to run VecScreen and classify matches as: TRUE_BIOLOGICAL, TRUE_ARTIFICIAL, FALSE_BIOLOGICAL, FALSE_AMR, etc. ([Fig. 1](#)). The first part identifies the candidate sequences and presents them as a FASTA formatted file. The first part is separate because it is used only in retrospective searches for contamination. In prospective analysis of database submissions, the submission itself contains the candidate sequences in FASTA format. The central algorithmic idea of the first part is to use the UniVec entries as blastn queries against the database being searched for contamination, with blastn parameters set so that any sequence that might have a VecScreen match is reported, to avoid false negatives.

The second part of the VecScreen_plus_taxonomy pipeline runs VecScreen, and organizes taxonomy and AMR information into a tab-delimited file with one VecScreen match per line, aided by the NCBI toolkit program `srcchk`. The third part is a program called `compare_vector_matches_wtaxa.pl` that does the actual classification. We implemented multiple versions of the third part varying in how much detail is included in input and output. A terse version of `compare_vector_matches_wtaxa.pl` was integrated into NCBI’s software system to evaluate new submissions. Since April 2017,

Fig. 1. Summary of the logic of the pipelines that use VecScreen_plus_taxonomy for both retrospective and prospective analysis. Retrospective analysis of existing GenBank sequences starts on the upper left. We use UniVec sequence as BLASTn queries to identify potentially contaminated sequences in GenBank and we apply syntactic filters based on Entrez terms early in the process to reduce the number of sequences to be considered. Prospective analysis of incoming submissions to GenBank starts on the upper right. See additional commentary in the text

NCBI personnel have been using the output while evaluating sequence submissions for possible vector contamination. Full instructions on how to invoke VecScreen_plus_taxonomy and examples can be found in the README file distributed with the software.

Our initial usage of VecScreen_plus_taxonomy has been to identify and to correct contaminated sequences in GenBank that are in the nonredundant (nr) nucleotide database; nr is the default database used in nucleotide BLAST searches. The presence of vector contamination in nr leads to false positive or at least misleading blastn matches, confusing BLAST users. To date, we applied the three-stage VecScreen_plus_taxonomy retrospective pipeline four times on the full nr database. We used filters (Supplementary File S1) to reduce the rate of matches that had a high probability of being false contamination. The filters were based on either Entrez terms or NCBI taxonomy ids. The syntactic filters (Supplementary File S1) were suggested by two biologists expert in Entrez (IK-M and RM), based on empirical patterns of false-positive matches in early testing. As vector source data improves, it may be possible to remove many filters.

In each successive round of searching nr, we decreased (and plan to continue decreasing) the number or stringency of the filters, so that more VecScreen matches are fully evaluated in the VecScreen_plus_taxonomy retrospective pipeline. Because we instantiated fewer taxonomy filters than Entrez filters and because it was essential for production usage of VecScreen_plus_taxonomy to allow through sequences of the extremely common taxid 77133 (Uncultured bacteria), we did the experiments on removing the taxonomy filters first (Supplementary File S1). In the fourth round, we used no taxonomy filters at all. During these four rounds, all matches deemed to be true contamination or some other problem were corrected by one of the first five correction options described in the Materials and methods section.

To date, 8314 sequences in nr have been corrected via use of VecScreen_plus_taxonomy (Supplementary File S2). Among these

8314 sequences: 1) 6465 sequences were trimmed, 2) 1422 were moved to the synthetic division of GenBank, 3) 122 received another correction such as the taxid, 4) 272 were unverified and 5) 33 were suppressed. We plan to rerun the analysis multiple additional times until no filters, other than excluding sequences in the synthetic division of GenBank, are needed.

The dates when contaminated sequences were deposited (Supplementary File S2 and Fig. 2) show that some vector-contaminated sequences resided in nr for over 20 years. The number of contaminated sequences deposited in recent years (2015–2017) is low, but not zero, suggesting that submission screening procedures have improved, but are not yet perfect. The use of VecScreen_plus_taxonomy screening should improve vector contamination screening of GenBank submissions. Further improvement is possible if one could find the biological sources of more vectors and vector segments or conclude that they are artificial.

Most of the recent work on software for vector contamination concerns trimming, assuming that contaminating vector sequences are located at the ends of sequences. Software packages developed for the task of removing terminal vectors, such as SeqTrim (Falgueras *et al.*, 2010), TagCleaner (Schmieder *et al.*, 2010), Btrim (Kong, 2011), primarily use libraries of known vector sequences analogous to UniVec. Other packages, such as Figaro (White *et al.*, 2008), AlienTrimmer (Crisuolo and Brisse, 2013), Skewer (Jiang *et al.*, 2014), PEAT (Li *et al.*, 2015) and SeqPurge (Sturm *et al.*, 2016), primarily use statistical techniques to identify subsequences at the ends that are inferred to be vectors, and hence are trimmed. We observed a gross underrepresentation of terminal matches among the sequences that we corrected, suggesting that these end-trimming software packages have been collectively successful. Since we observed in initial tests many contaminated sequences in which the contamination was internal VecScreen matches, we did not qualitatively distinguish between

Fig. 2. The number of sequences we corrected, classified by the year in which each sequence was originally loaded into GenBank

internal and terminal matches in VecScreen_plus_taxonomy, other than the different score thresholds already used in VecScreen.

Our work suggests one open problem: What laboratory procedures lead naturally and predictably to vector contamination in an *internal* location, as defined by VecScreen? In some cases, it is likely that a sequence read with untrimmed vector on one end was assembled with a read that had the same vector sequence at the beginning. Additionally, the ligation steps used in cloning procedures can occasionally create chimeric sequences that have pieces of vector in the middle. However, these two mechanisms cannot explain all the examples we found of true internal vector contamination.

Acknowledgements

Thanks to Olga Blinkova, J. Rodney Brister, Vyacheslav Brover, Christiam Camacho, Mark Cavanaugh, Michael Feldgarden, Daniel Haft, Jonathan Kans, Thomas Madden, Terence Murphy, Kim Pruitt and Conrad Schoch for technical assistance. Thanks to an anonymous reviewer and to the Associate Editor for suggestions that improved the manuscript.

Funding

This research was supported by the Intramural Research of the National Institutes of Health, National Library of Medicine (NLM).

Conflict of Interest: none declared.

References

- Altschul,S. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
- Binns,M. (1993) Contamination of DNA database sequence entries with *Escherichia coli* insertion sequences. *Nucleic Acids Res.*, **21**, 779.
- Camacho,C. *et al.* (2009) BLAST+: architecture and applications. *BMC Bioinform.*, **10**, 421.
- Coffin,J.M. *et al.* (eds.) (1997) *Retrovirus*. Cold Spring Harbor Laboratory Press, NY.
- Coker,J.S., and Davies,E. (2004) Identifying adaptor contamination when mining DNA sequence data. *Biotechniques*, **37**, 194–198.
- Criscuolo,A., and Brisse,S. (2013) AlienTrimmer: a tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads. *Genomics*, **102**, 500–506.
- Falgueras,J. *et al.* (2010) SeqTrim: a high throughput pipeline for pre-processing any type of sequence read. *BMC Bioinform.*, **11**, 38.
- Jiang,H. *et al.* (2014) Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC Bioinformatics*, **15**, 182.
- Kim,J. *et al.* (2016) Vecuum: identification and filtration of false somatic variants caused by recombinant vector contamination. *Bioinformatics*, **32**, 3072–3080.
- Kong,Y. (2011) Btrim: a fast, lightweight adapter and quality trimming program for next-generation sequencing technologies. *Genomics*, **98**, 152–153.
- Lamperti,E.D. *et al.* (1992) Corruption of genomic databases with anomalous sequence. *Nucleic Acids Res.*, **20**, 2741–2747.
- Li,Y.-L. *et al.* (2015) PEAT: an intelligent and efficient paired-end sequencing adapter trimming algorithm. *BMC Bioinform.*, **16**, S2, S1.
- Lopez,R. *et al.* (1992) Database contamination. *Nature*, **355**, 211.
- Miller,C. *et al.* (1999) A RAPID algorithm for sequence database comparisons: application to the identification of vector contamination in the EMBL databases. *Bioinformatics*, **15**, 111–121.
- Savakis,C., and Doelz,R. (1993) Contamination of cDNA sequences in databases. *Science*, **259**, 1677–1678.
- Schmieder,R. *et al.* (2010) TagCleaner: identification and removal of tag sequences from genomic and metagenomics datasets. *BMC Bioinform.*, **11**, 341.
- Schmieder,R., Edwards,R., and Rodriguez-Valera,F. (2011) Fast identification and removal of sequence contamination from genomic and metagenomics datasets. *PLoS One*, **6**, e17288.
- Seluja,G.A. *et al.* (1999) Establishing a method of vector contamination identification in database sequences. *Bioinformatics*, **15**, 106–110.
- Sturm,M. *et al.* (2016) SeqPurge: highly-sensitive adapter trimming for paired-end NGS data. *BMC Bioinform.*, **17**, 2018.
- Völter,C. *et al.* (1998) A broad spectrum PCR method for the detection of polioviruses and avoidance of contamination by cloning vectors. *Dev. Biol. Stand.*, **94**, 137–142.
- White,J.R. *et al.* (2008) Figaro: a novel statistical method for vector removal. *Bioinformatics*, **24**, 462–467.
- White,O. *et al.* (1993) A quality control algorithm for DNA sequencing projects. *Nucleic Acids Res.*, **21**, 3829–3838.