DETECTING BACTERIAL AND HUMAN CONTAMINATION IN DRAFT PROTEIN SEQUENCES

BIOSM901 Dissertation

Elliott Parris

FACULTY OF MATHEMATICS AND PHYSICAL SCIENCES,
UNIVERSITY COLLEGE LONDON, GOWER STREET, WC1E 6BT, UK

ABSTRACT

Sequence contamination is currently accepted to be simply unavoidable, despite the increasing reliance of a wide breadth of scientific fields on biological data, which may consequently lead studies to erroneous conclusion. This study implements and optimises an automated BLAST-based contamination detection method, which classifies and separates eukaryote queries as contaminated based on alignment statistics, with a large, local bacterial and human database. Using an e-value threshold of 1e-5 and a conservative identity threshold of 70%, 15.75357746% ± 0.006374096% of sequences from an assembled Xenoturbella transcriptome containing 32377 sequences were concluded to be contaminated eukaryotes and automatically separated from those uncontaminated into a separate file. This figure may be upwards of 33% using the same 1e-5 e-value threshold but with a less conservative 50% identity threshold – a value more comparable with that of a more sophisticated marker gene contamination detection method, Kraken. This was interpreted of this is horizontal gene transfer (HGT) events, from bacteria to eukaryotes, has led to eukaryote alignment with bacteria, from the contaminant database, resulting in false classification and inferred percentage contamination of the dataset. Used in seclusion, current BLAST-based and marker gene contamination detection methods have distinct advantages and limitations of application. Used in combination, with strategic layering, a novel automated contamination detection method could be synthesised which functions on all species, does not falsely classify queries based on HGT events yet excels at separating contaminated from uncontaminated queries: a blend not yet achieved.

Glossary: Metagenomic, Horizontal Gene Transfer, BLAST, Xenoturbella, Marker Gene

TABLE OF CONTENTS

INTRODUCTION

WHAT IS CONTAMINATION AND HOW DOES IT OCCUR?

Any sequence containing one or more segments from a foreign origin is contaminated. Whether it is a contaminated genome or a transcriptome, both embody the inclusion of a foreign biological sequence segment in a nucleotide sequence. There are a number of different sources of contamination, which all occur either in the isolation, purification, propagation or cloning steps in the preparation of DNA/RNA material, and/or as a consequence of one of many possible lab contaminants during this or another stage. There are various different sources of contamination, which are listed in Box 1.

It is possible to reduce events where contamination may occur, but even with rigorous management and high lab standards, at this point, this hazard cannot be eliminated entirely. It is a problem that occurs more frequently than is conceded, which some phylogenetic studies stress when trying to propose alternate theories (Koenemann, Jenner, Hoenemann, Stemme, & von Reumont, 2010; Reumont, Meid, & Misof, 2011; Wägele et al., 2009). Contamination is able to arise in a sample at any of the points typically involved in DNA extraction and later sample preparation (Naccache et al., 2013). Countless metagenomic studies, which attempt to classify genetic material from a variety of different species in a mixed community, but wish to focus on eukaryote material sampled, find significant levels of bacterial material in the genomes they assemble and later try to classify (Kumar, Jones, Koutsovoulos, Clarke, & Blaxter, 2013). Although Kumar et al. (2013) propose this is related to prokaryotes being more gene-rich, and that exons are more difficult to assemble than introns, Schmieder & Edwards (2011) found 4% of contamination in a human genome to occur in exons, on average, which is more than would be expected with only 1.1% of the genome consisting of exons (Venter et al., 2001). The process of contamination occurring is evidently not understood and requires further study, alongside methods for minimising its effects, specifically on downstream analysis.

This material may be the prey of the eukaryote being sampled, or a parasitic, symbiotic or commensal organism. Bacteria of the genera Delftia, Pseudomonas, Ralstonia, Propionibacterium and Lactobacillus are common sources of contamination in studies that are not metagenomic, as well as those that are, and are less avoidable; even with high lab standards (Clingenpeel, 2012).

Box 1: Sources of DNA contamination

Sequences from cloning vectors used to clone, propagate and manipulate the nucleotide sequence
Sequences from oligonucleotides such as adapters, linkers and PCR primers all used in the cloning or amplification of the nucleotide sequence
Transposable elements found in the foreign genomes of the cloning host may insert into the draft genome or transcriptome before propagation
Impurities in the DNA or RNA preparation, from which the sequences obtained can be partly or fully derived from, such as: • Nucleic acids from different sources in the same organism, an example being the presence of mitochondrial nucleic acids when only cellular nucleic acids should be present • Foreign nucleic acids from organisms used as biological reagents in other preparatory stages of the draft nucleotide sequences other than in the cloning or propagation, such as in the isolation or purification of the draft genome or transcriptome • Foreign nucleic acids from bacterial contaminants • Foreign nucleic acids, either due to cross contamination from insufficient lab equipment cleaning standards or human contamination, which may or may not be the result of poor lab procedure

Adapted from "Contamination in Sequence Databases," (2013)

WHY IS CONTAMINATION DETECTION A PROBLEM?

There are a number of different consequences of nucleotide sequence contamination, detailed in Box 2, which through a similar disruption effect, on the further analyses of the sequences, all lead to a detriment in scientific progression. This may be achieved via the waste of effort, money, and inclusively, yet most importantly - the time of research teams.

Box 2: Example mechanisms which describe ways contamination is detrimental to scientific progression

1st Example	Time and effort wasted on meaningless analyses
Description	Similarity searches based on foreign segments, between the draft genome or transcriptome and species sequences from public databases, may produce false positive results. Reviewing these to determine their possible significance wastes time and effort.

2nd Example	False conclusions drawn about the biological significance of the sequence
Description	Not reviewing these potential false positives can generate misleading data for possible functions and evolutionary relationships. Placing too much trust in the data may thus lead to even greater wastes in time and effort.

3rd Example	Incorrect assembly of genes in a sequence
Description	Sequences contaminated with the same foreign sequence can be aligned via the shared foreign segment. This can lead to joining or grouping of unrelated sequence contigs and false clustering of annotations.

4th Example	Delay in the release of the sequence in a public database
Description	In failing to identify and remove segments of foreign origin before submission the time needed to process the submission is significantly increased, which delays the release of the sequence.

5th Example	Pollution of public databases with sequences baring contamination
Description	Sequences from public databases are used for many different types of analyses. If contaminated sequences are added to public databases they can confound subsequent analyses of any data sets that include the contaminated sequences.

Bioinformatics, the major field affected by this effect, is widely acknowledged as one of a number of growing fields in the life sciences (Brusic, 2007; Pautasso, 2012; Song, Yang, & Tang, 2013), showing increases in citations per publication, publications and proportion of scientific publications. As is implied when connecting the two prior sentences, contamination induced problems should be expected to worsen; this big problem in bioinformatics has been, and will continue to, affect a wider range of researchers.

Bioinformatics is an interdisciplinary research area at the boundary between the biological and computational sciences, with computational biology lying closer to the latter with its focus transitioning more-so to large-scale analysis of, albeit biological, data. To understand why this project is so interesting one need only understand the interest in being able to use biological data to infer and gain a better insight into historical events and potentially predict those to occur in the future. This new knowledge will continue to have profound impacts on agriculture, the environment, energy, healthcare, and biotechnology.

In the wider field of bioinformatics where specialist projects, such as those modelling horizontal gene transfer (HGT) events, are part of a larger effort to account for the evolution of species. This will increase our, as yet, poor ability to model for future evolution which would capitulate a massive advancement in healthcare and conservational success. When using as advanced algorithms as these, the ability to produce positive results is significantly encumbered by segments of foreign origin - more so than segments of unknown identity, present in lower quality genomes.

WHAT IS WRONG WITH THE CURRENT STATE?

It is important therefore that safeguards are developed to recognise contaminated nucleotide sequences, and if possible eliminate them to avoid their accumulation in both research and public databases. While the occurrence of contamination is inevitable and the perturbing effects on further analysis are active, automated methods to identify and eliminate contaminated sequences from datasets will always be justified.

The National Centre for Biotechnology Information's (NCBI's) Vecscreen can identify vector, adapter, linker and Polymerase Chain Reaction (PCR) primer contamination. Of the four classes of DNA contamination that can occur, detailed in Box 1, the two this method addresses are the simplest to identify; so much so that many methods can remove these

instances of foreign genomic material with high confidence (Chen, Lin, Wang, Wu, & Hwang, 2007). Methods that attempt to tackle the more problematic instances of contamination, impurities in the DNA or RNA preparation, will have the largest positive effect on researchers. Around ten years ago no standardised contamination detection method, aimed at controlling the quality of genome sequences by doing just this, was present (Batzoglou, Jaffe, & Stanley, 2002). There are now an abundance of such methods.

The earliest methods all assessed contamination identity based on the degree of alignment between the query sequences and a chosen database of sequences of possible contaminants. The basic local alignment search tool (BLAST) program (Altschul, Gish, Miller, Myers, & Lipman, 1990) is often used as the alignment tool. Kuiken & Korber (1998) is one of the earliest examples of contamination detection using BLAST.

It is able to identify whether contamination from a common lab strain has occurred, but not whether contamination from less common strains used in the same facility has occurred. This has continued in similar, more recent examples of BLAST-based contamination detection method (Clingenpeel, 2012; Leggett & Ramirez-Gonzalez, 2013; Schmieder & Edwards, 2011). It may be owed to the advantages: speed and minimisation of the exclusion of sequences from the dataset for further analyses, due to suspicion they are contaminated, but in fact just possess similar segments of the genome because of HGT. The automation of operation and separation of input datasets into output files, according to the results of the contamination analyses, allows immediate further analysis if the researcher desires it. This feature further increases the time taken from sequence sampling to sequence analysis and is employed by (Schmieder & Edwards, 2011).

A BLAST alignment to only a limited number of species prevents the method being used on metagenomic data, which has become an increasing deterrent as metagenomic studies have become more common. With metagenomic data, there could be sequence segments of foreign origin from any number of species in the environment, where sampling and DNA preparation took place.

PROJECT AIMS

The three aims of this work were: to implement an automated and optimised BLAST-based contamination detection method that is accessible to a more general scientific researcher, or one new to the field; to discuss the advantages and disadvantages of different classes of

contamination detection methods; and to use the first two aims to discuss strategies for combining methods for further, faster improvements to automated contamination detection methods.

METHOD OVERVIEW

This method is made available to operate as a standalone application, the reasoning for which is to reduce the time taken for the method to complete relative to if it operated remotely through the use of the Internet. Its aim is to identify and separate contaminated eukaryote queries from otherwise uncontaminated eukaryote datasets of protein sequences. It is advised for application on datasets composed of eukaryote sequences, with the exception of strictly human datasets, because the method assumes the sequences of foreign origin to largely be of bacterial or human origin. The method automatically identifies bacterial and human sequences in the dataset, which despite aligning well to the assumed contaminants may or may not be contaminated by sequences from another bacteria or a human, or any bacteria respectively.

The automated eukaryote query contamination detection method designed here uses e-values and identity thresholds, the percentage of identical amino acids between the query and a high-scoring pair following a BLAST alignment, to determine if a match is a possible contamination or not. This approach is based on the idea that looking for similar regions consists of grouping sequences that share some minimum sequence similarity over a proportionate minimum length. To make it more accessible, simple parameters are used that can easily be changed according to the requirements of a particular study.

If trying to maximise the size of the dataset being used to infer HGT, this method should currently be avoided as an automated tool. The reason being it may remove queries for HGT analysis that are expected to have a higher degree of similarity, which is a problem encountered in a recent HGT study (Crisp, Boschetti, Perry, Tunnacliffe, & Micklem, 2015).

Three new files are automatically created once the contamination analysis method is complete: one contains eukaryote queries from the dataset identified as likely being contaminated, the second contains, both or either, prokaryote and human queries, whose contamination identity should be checked using another method, and the third contains eukaryote queries that are not contaminated. The third, uncontaminated, file can be used immediately for further analyses.

METHOD

LOCAL SOFTWARE REQUIRED

Three software packages were required in the local directory for method implementation: Python 3.4.2, which is available from https://www.python.org/downloads/release/python-342/; Biopython 1.65, which is available from http://biopython.org/wiki/Download; and NCBI BLAST 2.2.30+, a local standalone BLAST not to be confused with the BLAST web interface, is available for download on the NCBI FTP site ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/, either as a guest or a registered user. Standard documentation is available for all by following links from the respective home pages. The homepage for access to the NCBI BLAST 2.2.30+ standard documentation is http://www.ncbi.nlm.nih.gov/.

BLAST SEARCH INPUT

This method was applied to an assembled Xenoturbella transcriptome dataset (unpublished), from which proteins were predicted. It contains 32377 sequences, each with an average length of 280 amino acids. This multiple sequence FASTA file was provided as an input for the BLAST search to compare and align with a database.

BLAST DATABASE

The database contains a total of 8,790,821 sequences; of which, 8,719,483 are bacterial and 71,338 are human protein sequences (as of 26/03/2015) were both downloaded separately from NCBI. Both sets of sequences were divided further into sub-directories when downloaded from NCBI, with each folder corresponding to a species, or sometimes a specific sample of a species, and containing one or more sequences in FASTA format. Each of these FASTA sequences was concatenated into a bacterial and a human multiple-sequence FASTA file. Prior to the bacterial multiple sequence FASTA file being made into a database using BLAST's make blast database tool, as was also done for the human sequences, one copy of FASTA files with duplicate ID's had to be removed from the subdirectories where they were present due to an error being raised. Once this was resolved the reduced set of bacterial sequences were concatenated as was done previously and a bacterial database was produced to accompany the human database. Both databases were

combined into a single Bacterial and Human Protein Database using BLAST's database alias tool.

PYTHON SCRIPT

Below is a description of the full python script, which can be found in the Appendix.

BLAST SEARCH

The module NCBI Command Line Protein Blast was imported from Biopython's Blast Applications. This is required to run the BLAST search, using the queries as an input, against the Bacterial and Human Database to output BLAST results into an XML file with a defined name. For each query and its respective BLAST result there may be a number of different alignments returned and for each of these alignments there may be a number of high-scoring pairs (HSPs), sometimes termed hits. Defined in the BLAST search description was that all hits with an e-value greater than $10e^{-5}$ were eliminated, and the maximum number of hits in each alignment was restricted to 3. The number of different alignments for each query was not restricted. The number of cores used to execute the search was 32.

PREPARING QUERIES AND RESULTS FOR CONTAMINATION FILTERING FOR-LOOP

Three empty lists were created to allow queries to be appended to based on their response to criteria embedded in a for-loop, designed to identify contaminated eukaryote queries and any prokaryote queries. The three lists were contaminated_sequences, uncontaminated_sequences and check_sequences. NCBIXML and SeqI0 were imported from Biopython's Blast and Biopython respectively to allow BLAST results and queries to be parsed into a usable format. Lists were created for the BLAST results and queries, in this instance, to allow iteration through them both. A dictionary was made out of the list of query records to allow a query to be associated with its respective BLAST results. This was done by: splitting the query title given in the blast record, forming a list from each of the sections, and finally using the first list element as a key to find the corresponding query sequence using the dictionary.

CONTAMINATED EUKARYOTE AND ANY PROKARYOTE QUERY FOR-LOOP FILTER

The contamination identity of each of the 32377 queries is determined based on the identity and e-values of the hits given in the BLAST results for each query. For each BLAST record in the BLAST results, if there were no hits given the query sequence was appended to the list uncontaminated_sequences and the next BLAST record was iterated to by the for-loop.

For each hit in an alignment for a BLAST record, if either: the percentage identity was more than or equal to 70 and the e-value was not 0; or, the percentage identity was more than or equal to 70, the e-value was equal to 0 but the query sequence was different from the hit sequence, then the query sequence was appended to the list contaminated_sequences. Alternatively, if either: the percentage identity was less than 70; or the percentage identity was more than or equal to 70, the e-value was equal to 0, the query sequence was the same as the hit sequence and there were no more alignments the query sequence was also appended to the list uncontaminated_sequences. The last collection of criteria represents an unlikely scenario avert to the others, where in this case the contamination identity of the query is not confirmed, as true or false. The query returning itself as a BLAST result from the database does not mean the query is contaminated, and due to there being no other hits available to prove otherwise the query is deemed uncontaminated – 'innocent until proven guilty'.

Once a query sequence has been appended to a list, based on the interaction of one of its hits with the contamination identity criteria above, break clauses are initiated to skip to the next BLAST record in the for loop. If the contamination identity has not been concluded, where the percentage identity of every hit in an alignment was more than or equal to 70, the e-value equal to 0, the query sequence was the same as the hit sequence and there were more alignments, the query must be from a prokaryote or human, both of which the method cannot produce reliable results for; in regards to their contamination identity. In this instance, the query sequence is appended to the list check_sequences, containing queries whose contamination identity should be manually checked by another means.

Creation of Three FASTA Files for Further Analysis

A FASTA file was created from the list contaminated_sequences, separating contaminated eukaryote sequences from other queries in the dataset contained in the original FASTA file. Another FASTA file was created from the list check_sequences, separating prokaryote sequences from other queries in the dataset. A third FASTA file was created from the list uncontamined_sequences, containing uncontaminated eukaryote queries.

RESULTS

PARAMETER OPTIMISATION OF THE METHOD

Optimisation was performed using a randomly generated sample of 150 Xenoturbella sequences, which closely represented the full Xenturbella dataset in terms of the proportions of sequences contaminated.

The e-value and maximum number of hits per alignment returned for each query by the BLAST alignment were optimised. Their effect on the number of contaminants identified and the total time required for the script used in the appendix to completion on a randomly generated dataset containing 150 of the original Xenoturbella dataset were used to come to the conclusions for both their values.

The e-value, for which the method took the shortest time to complete and identified 27 contaminated queries, as does the least restrictive version of the method, where the e-value is 10, was $1e^{-5}$ (Figure 1).

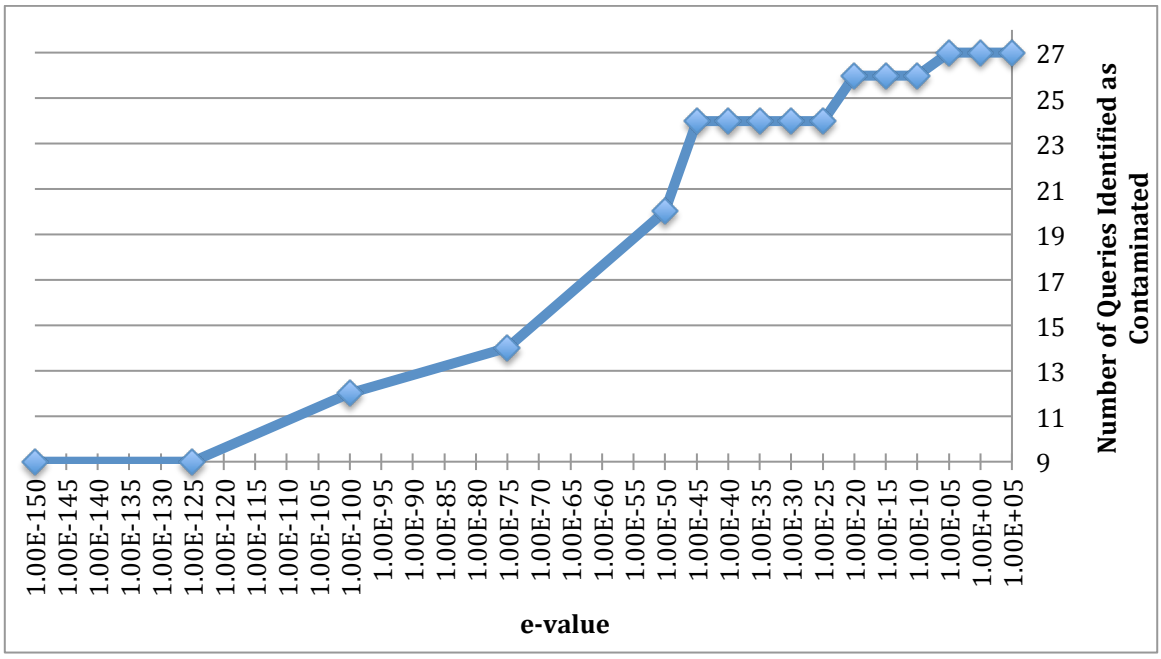

Figure 1: Effect of e-value on the Number of Queries Identified as Contaminated Eukaryotes. The e-values were changed in the python script including in the Appendix, which was executed on a random sample of 150 queries from the Xenoturbella dataset. The number of queries identified as contaminated shows an overall increase as the e-

value is increased (left to right). The rate of increase decreases until the number of contaminants identified as contaminated plateaus at 27 above an e-value of $1e^{-5}$.

Figure 2 shows how the time taken for the method to run to completion remains relatively constant until the e-value is increased to $\sim 1e^{-5}$. Between $\sim 1e^{-5}$ and $\sim 1e^{4}$ time increases moderately, and above $\sim 1e^{4}$ time increases rapidly.

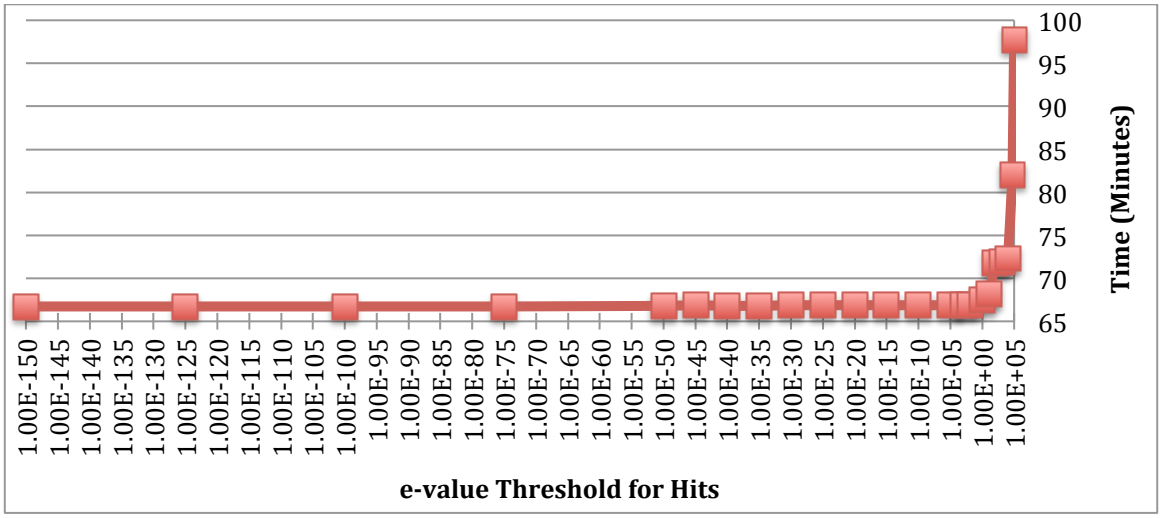

Figure 2: Effect of e-value on Time. The time taken for the script to run to completion was calculated using the user + system times from BioPython's 'time' tool. The e-values were changed in the python script including in the Appendix, which was executed on a random sample of 150 queries from the Xenoturbella dataset. The time taken for the script to run to completion is steady with an increasing e-value threshold (left to right) until $1e^{-5}$, after which point the rate of increase of time with increasing e-value increases.

Changing the maximum number of hits had no effect on the number of contaminated queries identified. In regards to the time taken for the method to complete, using three gave a marginally shorter completion time. The difference between using three as the parameter and one, the next fastest, was under 5 seconds for a process taking approximately 67 minutes to complete.

As for the percentage identity (50% - 90%), the time taken for the method to complete is shortest using 70 and above as the threshold. As the percentage identity threshold is increased there is a decrease in the number of queries identified as contaminants by the

respective method. The rate of decrease increases when changing the percentage identity from 50 to 70, but begins to decrease from 70 to 90.

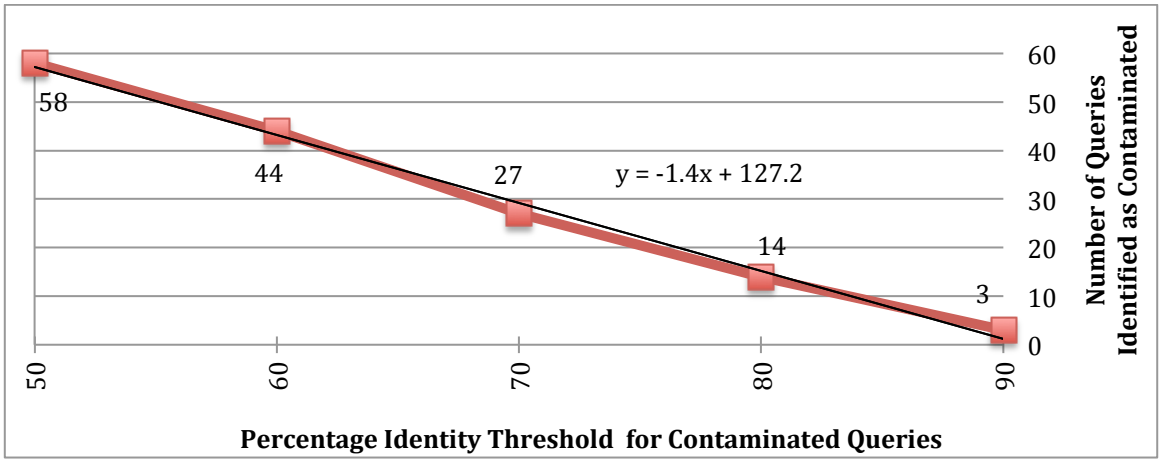

Figure 3: Effect of Percentage Identity Threshold on Number of Queries Identified as Contaminated Eukaryotes. The e-values were changed in the python script including in the Appendix, which was executed on a random sample of 150 queries from the Xenoturbella dataset. The number of queries identified as contaminated shows a steady decline as the percentage identity threshold is increased (left to right).

METHOD OUTPUT FROM XENOTURBELLA DATASET INPUT

Using an e-value threshold of $1e^{-5}$, a maximum number of three hits for an alignment for each query and a percentage identity threshold for hits of 70.0 the method was executed on the whole dataset. Although the actual time taken was 13 hours and 49 minutes using 32 cores, the total time taken if the script was run from only one thread in addition to the time the system needed to keep the job running, but not actually computing anything, was 20 hours and 45 minutes. Of the 32377 Xenoturbella sequences: 4943 were deemed to be contaminated eukaryotes, 26430 as uncontaminated eukaryotes and 4 as bacteria or human queries, which may or may not be contaminated. This method thus renders the dataset to have $15.75357746\% \pm 0.006374096\%$ of its queries contaminated.

The results from executing the method with a 70% identity threshold on the randomly generated sample of 150 queries of the Xenoturbella dataset inferred 27 contaminated queries. 24 would have been expected, if the sample were perfectly representative of the full Xenoturbella dataset, which was found to have approximately 15.75% of queries being contaminated. The sample has approximately 12.5% more contaminated species than is representative of the full dataset.

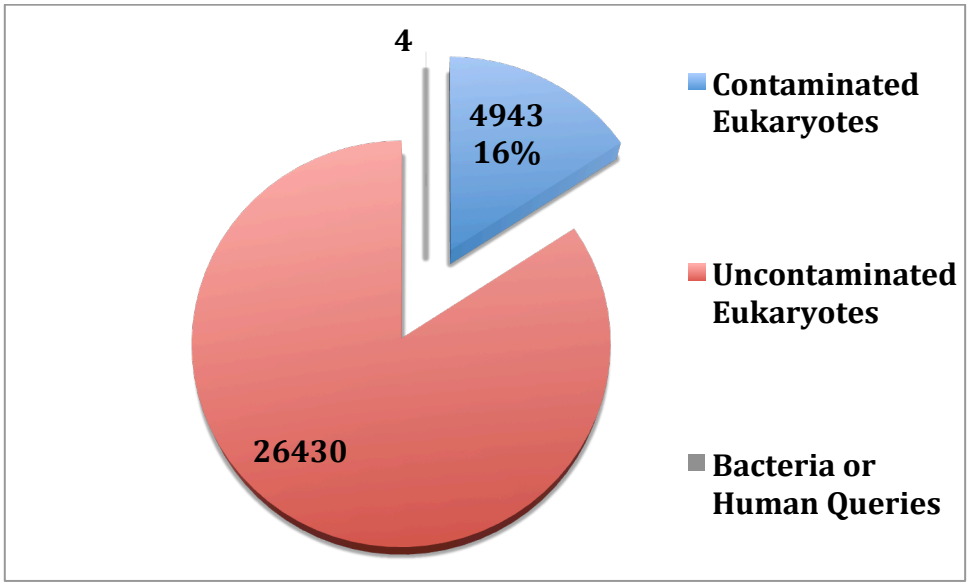

Figure 4: The Contamination Identity of Queries in the Xenoturbella Dataset. Each query was classified according to the following criteria, where the presence of a '!' before signifies 'Not': A = % Identity ≥ 70.0; B = e-value == 0.0; C = Query Sequence == Hit Sequence; D = There are more alignments. Queries were identified as: Contaminated Eukaryotes (blue) if $A \cap !B$ or $A \cap B \cap !C$; Uncontaminated eukaryotes (red) if $!A$ or $A \cap B \cap C \cap !D$; or Bacteria or human queries if $A \cap B \cap C \cap D$. Of the 32377 sequences: 4943 (15.75357746%) were identified as contaminated eukaryotes; 26430 (84.23367435%) as uncontaminated eukaryotes; and 4 (0.012748191%) as bacteria or human sequences.

DISCUSSION

INTERPRETATION OF RESULTS

Through inputting the actual percentage identity threshold, which gave an actual number of contaminated queries, it was concluded using 70 as the identity threshold gives the most conservative estimate of contamination, identifying 2 less contaminated queries than would be expected considering the numbers given from all five identity thresholds. This justified its use for later analysis in this instance. However, without comparison of this methods performance with that of another, with both assessing the contamination identity of the same dataset, or a simulation of contamination to show how accurate the method is; the user should consider changing this threshold to what they deem most appropriate.

When the highly sophisticated Kraken contamination detection method (Wood & Salzberg, 2014) was executed on the same Xenoturbella dataset by another member of the Dessimoz Lab (Stephen Müller, unpublished work) up to 40% (potential) contamination was identified. Kraken uses short sequences as markers of contamination, which are termed 'marker' genes, as opposed to looking for complete sequences, as do BLAST-based methods such as this. These marker genes are highly conserved between species, which allows the method to create a reference genome tree, and classify input queries according to the alignment of their homologous marker gene to one indexed to a location in the tree functioning as a database.

Assuming from the results that samples classify approximately 12.5% more queries to be contaminated, despite 58 out of 150 queries being identified as contaminated using a 50% identity threshold, which accounts for 38.67% of the sample queries, 32.87% of the queries in the whole 32377 sequence dataset would be expected to be contaminated; if also using a 50% identity threshold.

Despite the heightened complexity of the Kraken contamination detection method, there is no guarantee the result given using their method on the Xenoturbella dataset, which estimates a slightly higher percentage of queries are contaminated, is correct. Even if query sequences are classified as being distinct from the Xenoturbella, this classification, as an indicator of contamination, is based on short reads and the degree of contamination at this reference point may be unrepresentative of the rest of the sequence, which may lead to false positive classification of the query as being contaminated. BLAST-based contamination methods, although slower and less sophisticated, make assumptions based on the whole query sequence. Although, this is more susceptible to falsely classifying queries as contaminated from detecting HGT events.

Review of Long-Read Alignment Contamination Detection Methods

DeconSeq

Sequence alignment currently underlies all methods of contamination identification and removal. Schmieder & Edwards (2011) averted from using the BLAST alignment program, and opted for BWA-SW (Li & Durbin, 2010), a faster method which also aligns long-length sequences, for their method - DeconSeq. Method speed was addressed early in the design of this project, and is something that has been improved upon by using a local BLAST search

as opposed to operating the alignment program remotely, as done in similarly designed, earlier methods (Turnbaugh et al., 2009; Willner et al., 2009).

DeconSeq uses coverage and, similarly, identity thresholds, to determine if a match is a possible contamination or not. This approach is based on the idea that looking for similar regions consists of grouping sequences that share some minimum sequence similarity over a specified minimum length. The use of e-values, which are based on probabilities, will indirectly confer a minimum length in this method also.

Single Cell Data Decontamination

The method developed by Clingenpeel (2012) is different in that it assumes contamination to commonly occur via a selection of bacteria, ignoring human contamination. Its restricted database confers an increase in the speed of analysis, but renders the method unable to be applied to metagenomic datasets. Clingenpeel (2012) identify another limitation of their method which is that HGT events may be incorrectly identified as contamination.

Kontaminant

An algorithm known as Kontaminant employs a similar method to this whereby it looks for overlap between draft nucleotide sequences and contaminants (Leggett & Ramirez-Gonzalez, 2013), as is done in this method via BLAST, contrary to Kontaminant. Kontaminant returns the percentage of reads possessing one or more kmer hits, which are small, and in this case 21 base-long, segments aligned with the query. These are analogous to HSPS returned by BLAST alignments. It also returns the percentage of the highest matching contaminant genome, from the database, which is covered by the query sequence.

These two additional functions, particularly the last one, allow the function of a novel strategy to bypass the false classification of HGT events as query contamination. If the percentage coverage of the contaminant by kmers aligned with the query does not reach a certain threshold it is indicative these are shared between bacterial species, maybe as a consequence of HGT events, is not actually contamination and are thus ignored. The user may also specify which of the reference contaminants they want to compare overlap with, if they desire. Because Kontaminant assumes contamination to commonly occur via a selection of bacteria, the method also has a small database and is fast, but cannot be applied to metagenomic datasets.

Limitations of this method

The method assumes the genomes sourced for the combined bacterial and human database lack contamination (Benson et al., 2012), although this is unlikely to be wholly true. Having multiple sequences for many of the bacteria in the database mitigates this potential limitation. Although the likelihood of bacterial database sequence contamination impeding the method is extremely low, it is acknowledged it may remain a problem, particularly for rarely sequenced genomes.

Crisp et al. (2015) and Clingenpeel (2012) both raise the issue that queries labelled as contaminated may be an artefact of the contamination detection method, due to a HGT occurring between a query sequence segment and a sequence from the database of potential contaminants. Crisp et al. (2015) mention how the use of multiple closely related species, in their HGT classification method, to define ortholog groups, inherent in their definition of class B HGT, renders these less prone to be contaminants than Class C HGT. Similarly they mention the impact of contamination on increasing false positive results inferring HGT events using high quality databases. This effect, which can be slightly avoided using multiple closely related species to define ortholog groups, would be even more pronounced in lower quality databases. Limiting the number of queries labelled as contaminated using this method, but maintaining the degree of contamination detection represents a key goal of further study.

'Marker' Gene Contamination Detection Methods

Early Methods

An increasing number of genome annotations over time made it easier to define homologous sets of genes present in different species. Some of the functions these homologous genes code for are highly derived and thus can be present in large clades; of which, there are examples that can be found in almost all bacteria (Parks, Imelfort, Skennerton, Hugenholtz, & Tyson, 2014). Those present in only one copy, and no more, offer the opportunity for the detection of contamination. If there are multiple copies identified, of what is a known single-copy marker gene, this is a clear indication of contamination of the query. Specifically, the foreign sequence is a marker gene from the contaminant species.

Liu, Gibbons, Ghodsi, Treangen, & Pop (2011) and Segata et al. (2012) were two of the earliest studies outputting methods, MetaPhyler and MetaPhlAn respectively, that utilise marker genes, chiefly the numbers of these present in a query, for the identification of bacterial and archaic microbes. There are a number of more recent metagenomic studies,

which have used marker genes present in multiple copies, known to be single-copy marker genes, to infer contamination and genome completion after assembly (Albertsen et al., 2013; Soo et al., 2014).

Novel Methods

The studies producing Kraken (Wood & Salzberg, 2014), the marker gene method discussed earlier, and CheckM (Parks et al., 2014) are the first to fully utilise the potential contamination detection applications of marker genes. They have both benefited from the increasing volume of studies aiming to infer HGT events; for which its decreased mistaking of HGT events as contamination offers it an advantage over more conventional BLAST-based methods.

The less sophisticated example of the two marker gene contamination detection methods, CheckM (Parks et al., 2014) still has a much larger range of marker genes than MetaPhyler and MetaPhlAn. Kraken's is larger than CheckM's by a similar degree. Parks et al. (2014) found that 36% of marker genes were collocated in one or more genomes and could thus be grouped into marker sets. The method also uses a database of much fewer bacterial genomes than this method, however theirs are all located at the nodes of large bacterial clades, and the nodes linking each of those clades. This allows their method to create a reference genome tree; the nodes of which are indexed with their corresponding marker sets. Rather than check for more than one copy of the same single marker gene in each query, their method chooses different marker sets to compare each query to based on the position of the query genome in the reference genome tree.

It is unclear from Parks et al. (2014) how the position of each query in the reference genome tree is determined. However, considering the smaller database used and the contigs being aligned, which are shorter in length than when using this method, it is safe to assume this to be faster than BLAST-based contamination detection methods. One of the other positives is that it allows significantly more reliable contamination detection in bacterial sequences and, less impressively, human sequences than BLAST-based methods, maybe with the exception of Schmieder & Edwards (2011).

Another positive of this method is that for marker genes chosen wisely, the limitation of false classification of HGT events as contamination can be avoided using this method. In practice this would involve choosing which marker gene, found in all of the dataset's queries, to index to the reference genome tree based on: the likelihood to be involved in

HGT events; in addition to other conventional selection criteria, such as the extent of diversity that the marker gene encompasses (Parks et al., 2014).

BLAST-based methods can be executed on datasets of greater size, which is a limitation of marker gene based methods because they rely on one, of a limited number of marker genes (Sharon & Banfield, 2013), being present in all the query sequences. Another key advantage of using automation with conventional BLAST-based contamination detection methods, as implemented here, as opposed to methods using marker genes is simple, but easily overlooked. Although all contamination detection methods are approximations of the extent sequences of foreign origin are present, a marker gene represents only one of all the genes in a query sequence, and this small sample is used to infer the contamination identity of the rest of the query sequence. This is therefore an approximation of an approximation.

FURTHER STUDY

POTENTIAL FOR COMBINATION OF BLAST-BASED AND MARKER GENE CONTAMINATION DETECTION METHODS

Although Parks et al. (2014) only implemented this method on bacterial and archaic sequences, the same principles could be applied to produce eukaryote marker genes and marker sets. There is scope for this to function as a more stringent layer of contamination detection to be implemented on top of the current method. An extra filtering would only need to be applied for queries labelled as contaminated, to check if the segments of foreign origin arose naturally in a HGT event, or whether their presence hinders the faithful representation of the nucleotide sequence before sample preparation. This would be expected to decrease the false positive classification of queries as contaminated, as a consequence of a HGT, which are not so.

Contaminated sequences could then be classified based on their contamination assessment by both the first eukaryote-query contamination detection layer, used in this study, and the eukaryote marker gene layer. Similar to the classification method used in Crisp et al. (2015): queries passing only the first layer could be labelled as having Class-B contamination and those passing both – Class-A contamination. New multiple sequence files can be automatically created. One would contain all contaminated queries, with another containing all uncontaminated queries – the same as is done in this study. An additional two files could

be created, only with this improved method, separating queries according to whether or not they have Class-A contamination. This would output a file with a larger amount of certainty as to the contamination identity of the queries, in addition to a larger number of uncontaminated queries, proportional to the number of queries with Class-B contamination, to work with.

Potential for Combination of BLAST-Based Contamination Detection Methods Using Smaller and Larger Databases

The query contamination analysis by Clingenpeel (2012) is faster than this method. In BLAST-based methods speed is largely dependent on the speed of the BLAST search, and inversely proportional to the size of the database query sequences are aligned to. This could be used in combination with the method designed in this study. The overall speed would be increased at no loss of queries identified as contaminants, if the method by Clingenpeel (2012) was executed first. The degree to which layering of the contamination detections method in this order would improve the speed of the overall process would be contingent on how many queries were identified as contaminants before the second, and more stringent layer of filtering, was executed to identify the remaining contaminated queries.

Other Potential Improvements in Further Study

Very little can be addressed in terms of the robustness of the contamination identification other than using a database based on the most up to date bacterial and human sequences. Rather than executing the BLAST search against a database using human genomes, downloaded from NCBI, human genomes from the Human Genome Project, with a higher genome quality, could be used in the database instead.

Conclusion

Does contamination occur at a higher rate in genomes sites or segments with particular characteristics or functionality? Despite the time and effort that goes into minimising the numerous, detrimental, downstream effects of contamination, in addition to the effects themselves, it is surprising this question has not become more central, as has mutational analysis in healthcare and biochemical engineering. In regards to contamination detection methods, that implemented in this study represents one of the few able to detect both human and bacterial contamination, and offers flexibility for the user to tailor the simple parameters

for their research requirements. The BLAST-based contamination detection method was effective in identifying, using the most conservative identity threshold of 70%, and separating, approximately 16% of contaminated eukaryote queries from the large Xenoturbella dataset. This figure may be upwards of 33% using the same $1e^{-5}$ e-value threshold but with a less conservative 50% identity threshold – a value more comparable with that of the more novel marker gene contamination detection methods.

Used in isolation, BLAST-based and marker gene contamination detection methods have distinctive advantages and disadvantages, with the latter potentially having less. A multi-layered method combining the two would no longer require a file created for queries requiring manual checking because it would function not just on eukaryotic sequences, as this method currently does (excluding humans), or just on bacterial and archaic sequences (Crisp et al., 2015), but all sequences. An automated contamination detection method, which is thorough, yet does not automatically remove queries based on HGT events and also functions on all species would be the gold standard. A goal that must be strived for with the increasing reliance of researchers, from all manners of disciplines, on, somewhat contaminated biological data.

ACKNOWLEDGEMENTS

I would like to thank Stephen Müller, for developing the project and the guidance creating the script for the method, Christophe Dessimoz for the opportunity, continued guidance and support; along with the rest of the Dessimoz Lab.

REFERENCES

Albertsen, M., Hugenholtz, P., Skarshewski, A., Nielsen, K. L., Tyson, G. W., & Nielsen, P. H. (2013). Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nature Biotechnology*, *31*(6), 533–8. doi:10.1038/nbt.2579

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–10. doi:10.1016/S0022-2836(05)80360-2

Batzoglou, S., Jaffe, D., & Stanley, K. (2002). ARACHNE: a whole-genome shotgun assembler. *Genome, (11) 1*

Benson, D. A., Karsch-Mizrachi, I., Clark, K., Lipman, D. J., Ostell, J., & Sayers, E. W. (2012). GenBank. *Nucleic Acids Research*, *40*(Database issue), D48–53. doi:10.1093/nar/gkr1202

Brusic, V. (2007). The growth of bioinformatics. *Briefings in Bioinformatics*, *8*(2), 69–70. doi:10.1093/bib/bbm008

Chen, Y.-A., Lin, C.-C., Wang, C.-D., Wu, H.-B., & Hwang, P.-I. (2007). An optimized procedure greatly improves EST vector contamination removal. *BMC Genomics*, *8*(1), 416. doi:10.1186/1471-2164-8-416

Clingenpeel, S. (2012). *Single Cell Data Decontamination*.

Contamination in Sequence Databases. (2013). Retrieved from http://www.ncbi.nlm.nih.gov/tools/vecscreen/contam/

Crisp, A., Boschetti, C., Perry, M., Tunnacliffe, A., & Micklem, G. (2015). Expression of multiple horizontally acquired genes is a hallmark of both vertebrate and invertebrate genomes. *Genome Biology*, *16*(1), 50. doi:10.1186/s13059-015-0607-3

Koenemann, S., Jenner, R. a, Hoenemann, M., Stemme, T., & von Reumont, B. M. (2010). Arthropod phylogeny revisited, with a focus on crustacean relationships. *Arthropod Structure & Development*, *39*(2-3), 88–110. doi:10.1016/j.asd.2009.10.003

Kuiken, C., & Korber, B. (1998). Sequence Quality Control, 80–90.

Kumar, S., Jones, M., Koutsovoulos, G., Clarke, M., & Blaxter, M. (2013). Blobology: exploring raw genome data for contaminants, symbionts and parasites using taxon-annotated GC-coverage plots. *Frontiers in Genetics*, *4*, 237. doi:10.3389/fgene.2013.00237

Leggett, R., & Ramirez-Gonzalez, R. (2013). Sequencing quality assessment tools to enable data-driven informatics for high throughput genomics. *Frontiers in …*.

Li, H., & Durbin, R. (2010). Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, *26*(5), 589–95. doi:10.1093/bioinformatics/btp698

Liu, B., Gibbons, T., Ghodsi, M., Treangen, T., & Pop, M. (2011). Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. *Genome Biology*.

Naccache, S. N., Greninger, A. L., Lee, D., Coffey, L. L., Phan, T., Rein-Weston, A., … Chiu, C. Y. (2013). The perils of pathogen discovery: origin of a novel parvovirus-like hybrid genome traced to nucleic acid extraction spin columns. *Journal of Virology*, *87*(22), 11966–77. doi:10.1128/JVI.02323-13

Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P., & Tyson, G. W. (2014). CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. doi:10.7287/peerj.preprints.554v1

Pautasso, M. (2012). Publication Growth in Biological Sub-Fields: Patterns, Predictability and Sustainability. *Sustainability*, *4*(12), 3234–3247. doi:10.3390/su4123234

Reumont, B. M. Von, Meid, S., & Misof, B. (2011). Wide Spectra of Quality Control. In I. Akyar (Ed.), *Wide Spectra of Quality Control*. InTech.

Schmieder, R., & Edwards, R. (2011). Fast Identification and Removal of Sequence Contamination from Genomic and Metagenomic Datasets. *PLoS ONE*, *6*(3).

Segata, N., Waldron, L., Ballarini, A., Narasimhan, V., Jousson, O., & Huttenhower, C. (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature …*, *9*(8), 811–4. doi:10.1038/nmeth.2066

Sharon, I., & Banfield, J. F. (2013). Microbiology. Genomes from metagenomics. *Science (New York, N.Y.)*, *342*(6162), 1057–8. doi:10.1126/science.1247023

Song, M., Yang, C. C., & Tang, X. (2013). Detecting evolution of bioinformatics with a content and co-authorship analysis. *SpringerPlus*, *2*(1), 186. doi:10.1186/2193-1801-2-186

Soo, R. M., Skennerton, C. T., Sekiguchi, Y., Imelfort, M., Paech, S. J., Dennis, P. G., … Hugenholtz, P. (2014). An expanded genomic representation of the phylum cyanobacteria. *Genome Biology and Evolution*, *6*(5), 1031–45. doi:10.1093/gbe/evu073

Turnbaugh, P. J., Hamady, M., Yatsunenko, T., Cantarel, B. L., Duncan, A., Ley, R. E., … Gordon, J. I. (2009). A core gut microbiome in obese and lean twins. *Nature*, *457*(7228), 480–4. doi:10.1038/nature07540

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., … Zhu, X. (2001). The sequence of the human genome. *Science (New York, N.Y.)*, *291*(5507), 1304–51. doi:10.1126/science.1058040

Wägele, J. W., Letsch, H., Klussmann-Kolb, A., Mayer, C., Misof, B., & Wägele, H. (2009). Phylogenetic support values are not necessarily informative: the case of the Serialia hypothesis (a mollusk phylogeny). *Frontiers in Zoology*, *6*, 12. doi:10.1186/1742-9994-6-12

Willner, D., Furlan, M., Haynes, M., Schmieder, R., Angly, F. E., Silva, J., … Rohwer, F. (2009). Metagenomic analysis of respiratory tract DNA viral communities in cystic fibrosis and non-cystic fibrosis individuals. *PloS One*, *4*(10), e7370. doi:10.1371/journal.pone.0007370

Wood, D. E., & Salzberg, S. L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, *15*(3), R46. doi:10.1186/gb-2014-15-3-r46

APPENDIX

PYTHON SCRIPT

Parts the user will need to change and may want to change, according to their needs and preferences, are in bold and bold with italics respectively.

```
#####Part 1 - BLAST eukaryotic queries against a local human and bacterial database
#Module imported
from Bio.Blast.Applications import NcbiblastpCommandline #Needed to run the BLAST search using the queries as the input
#Directories
name = "Xenoturbella_24JAN_preclast.fa"
data_base = "bacterial_and_humanDB.faa"
out_file = "Xenoturbellavs2and9606.xml"
#BLAST search - 32 threads, max of 3 hits per query
comline = NcbiblastpCommandline(query=name, db=data_base, evalue= 1e-5, max_target_seqs=3, outfmt=5, out=out_file, num_threads=32)
stdout, stderr = comline()
#####Part 2 - Parse BLAST Results and Queries and create empty lists to append queries to based on their response to criteria
#Modules imported
from Bio.Blast import NCBIXML #Needed to parse the BLAST output into a usable format
from Bio import SeqIO #Needed to parse the queries
#Empty Lists to append query sequences too based on how their BLAST results respond to contaminated criteria
contaminated_sequences = []
uncontaminated_sequences = []
check_sequences = []
#Query record list and dictionary
query_records = list(SeqIO.parse(name, "fasta")) #list created to allow iteration through query records
query_record_dict = SeqIO.to_dict(query_records)
#Blast record list
result_handle = open(out_file)
blast_records = list(NCBIXML.parse(result_handle))
#####Part 3 - Filter out Contaminated Queries using a For-Loop to iterate through parsed BLAST Results and Queries
for blast_record in blast_records: #for each blast record in the BLAST results
   split_query_title = list((blast_record.query).split()) # making a list out of the words in the query title for the blast record
   if len(blast_record.alignments) == 0: #if there are no results given
      uncontaminated_sequences.append(query_record_dict[split_query_title[0]]) #appends the query record from the dictionary with the key given from the first word of the query title for this blast record
```

```
        continue # continue iterating through the for-loop
    concluded = False # at this point in the for loop the contamination identity is not yet concluded
    for alignment in blast_record.alignments:
        if concluded: #if this query has been appended to one of the lists
            break #skip to the next blast record
        for hsp in alignment.hsps: #for each high scoring pair in this blast record
            if concluded: #if this query has been appended to one of the lists
                break #skip to the next blast record
            A=hsp.identities*100.0/(len(hsp.sbjct)) >= 70.0 #identity>=70
            B=hsp.expect == 0.0 #evalue=0
            C=hsp.query == hsp.sbjct #'draft' query sequence=hit sequence. Therefore it is not a draft sequence
            D=len(blast_record.alignments) > 1 #there are more alignments
            if (A) and ((not (B)) or (not (C))) or (A) and (not((B) and (C))): #if AnBn!C or An!B
                ''' If % identity>=70, evalue=0 but query sequence different from hit sequence:
                or % identity>=70 and evalue is not 0:'''
                contaminated_sequences.append(query_record_dict[split_query_title[0]]) #appends the query record from the dictionary with the key given from the first word of the query title for this blast record
                concluded = True #at this point in the for loop the contamination identity is concluded
            elif (not (A)) or (not(D)): #if A! or AnBnCn!D
                ''' If % identity<70:
                or % identity >=70, evalue=0, query sequence=hit sequence and there are no more alignments:'''
                uncontaminated_sequences.append(query_record_dict[split_query_title[0]]) #appends the query record from the dictionary with the key given from the first word of the query title for this blast record
                concluded = True #at this point in the for loop the contamination identity is concluded
        if concluded: #if this query has been appended to one of the lists
            break #skip to the next blast record
    if not concluded:  #if a query has not been appended to one of the lists after iterating through all hsps given, assume it is a bacterial or human sequence
        check_sequences.append(query_record_dict[split_query_title[0]])  #appends the query record from the dictionary with the key given from the first word of the query title for this blast record
#####Part 4 - Write lists created to three distinct new fasta files seperating queries according to contamination status
SeqIO.write(contaminated_sequences, "contaminated_xenoturbella_queries.fa", "fasta")
SeqIO.write(uncontaminated_sequences, "xenoturbella_queries.fa", "fasta")
SeqIO.write(check_sequences, "check_xenoturbella_queries.fa", "fasta")
```