Contaminations in genomic sequences

Pavankumar Videm [a], Dominic Rose [a,b]

[a] *Bioinformatics Group, Department of Computer Science, University of Freiburg,
Georges-Köhler-Allee 106, D-79110 Freiburg, Germany.*

[b] *Supervisor:* `dominic@bioinf.uni-freiburg.de`

Manuscript *18 August 2011*

Abstract

Despite continued advances in whole genome sequencing techniques and the development of powerful assembly algorithms, newly sequenced genomes still often suffer from contaminations during the sequencing process. The most common sources of contamination are accessory DNAs deliberately attached to the DNA/RNA under investigation, including vectors, adapters, linkers and PCR primers. However, there are also unintended events, e.g. caused by transposon activity or simply impurities, leading to contaminated genomic sequences. These may then result in missassemblies of genomic sequences, meaningless analyses and potentially erroneous conclusions. However, no one knows to which extent publicly available genomes are contaminated.

To encompass this unsatisfying situation we therefore plan to develop a comparative genomics approach to broadly identify contaminations in available genomic sequences. However there exist some tools those can find contaminations from adapters or from vectors alone. Here we present an approach based on machine learning to distinguish between contaminated and non-contaminated sequences instead of finding vector contaminations or adapter contaminations alone. As for now no such tool available, our approach would be foremost and showed promising results on different datasets.

Key words: contamination, comparative genomics, vectors

Introduction

Here, we explain what is contamination, consequences of contaminations and our approach to distinguish contaminated sequences from non-contaminated sequences. A genome contamination is a subsequence(s) on the genome of a species which does not represent the actual genetic information relevant to that species. So contaminations are foreign sequences. Mostly, contaminations are from vectors, adapters primers and linkers. While sequencing genomes people deliberately use these vectors, adapters etc. for cloning and liking of sequences and leave them unintentionally in final genome assemblies and contaminate them.

Contamination sources

Fig.1 explains one of the generic scenario of such contamination source. While sequencing genomes chromosomes are split into small DNA fragments. Then these DNA fragments are placed into vectors. To clone DNA these vectors are then shocked into bacteria. These bacterias multiply, which in turn makes copies of the vectors holding DNA fragments. Then these cloned vectors are recovered from the bacteria. Primers are attached itself to a complementary sequence on the vector to build new strands of DNA. In this process

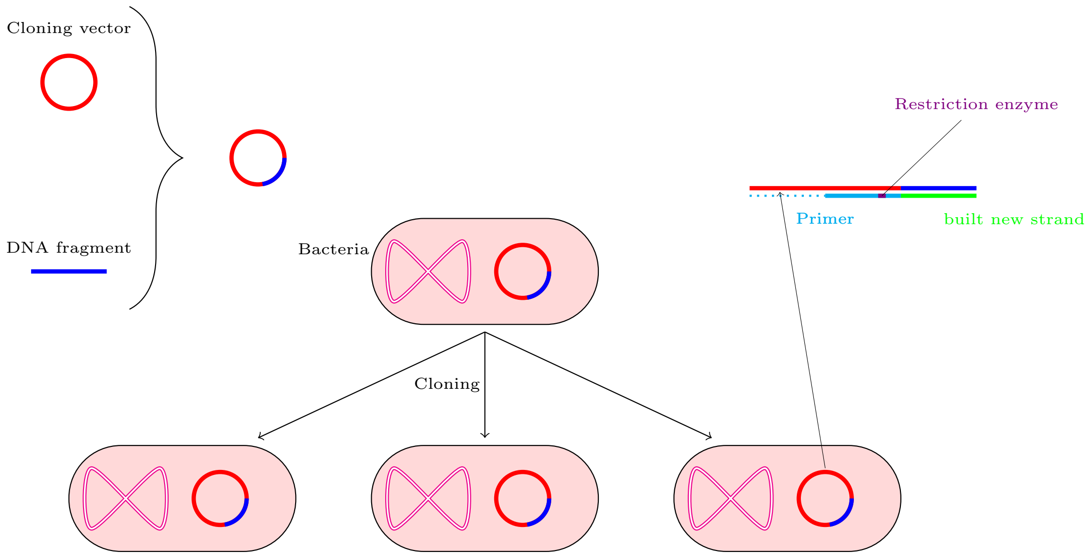

Fig. 1. To clone DNA, DNA fragments are combined with cloning vectors and then shocked into bacteria. Then bacteria reproduce several copies of itself and also vectors. All vectors are recovered from bacteria and continue to sequence whole genome. After this, unable to find and remove the vectors or other sequencing artifacts lead to contaminated genomes.

incorrect placement of restriction enzyme or recheck and removal of vector content in the final assembly or some other artifacts result in contaminations.

Consequences

There are several consequences of genome contaminations

- Now a days, number of public sequence databases available on the web became the main sources of information for analysts. Depositing contaminated sequences on public databases can be literally polluting the databases.
- Working with data containing obscure contaminations can be precarious. One may draw erroneous conclusions about the biological significance of the sequence. Hence time and effort wasted on meaningless analysis.
- Identification and removal of vectors and preprocessing of sequences before submitting to databases is essential. But this causes delay in release of sequences.

People worked before on identification and removal of contaminations, but their approaches are dependent on type/source of contamination. For example tools have been developed to find the vector (1; 2) or adapter (3) contaminations. The theme of our project is to make a model that can recognize contaminated and non-contaminated sequences irrespective of sources of contaminations. To develop such a tool, machine learning approaches are used to train a model that can discriminate contaminated from non-contaminated sequences.

Materials and Methods

We acquired the genome assemblies of 58 different metazoans from UCSC Genome Browser (4) and a total of 4696 unique vector sequences from NCBI *UniVec* (5) and Stanford's *Vectordb* (6) databases. The final model should be made out of contaminated and non-contaminated sets of sequences and should be able to distinguish between the two sets. The idea of obtaining these two sets of sequence is straight forward. Align vectors against genomes, then all aligned sub sequences on the genomes can be contaminated as they represent genetic information of vectors and remaining can be valid non-contaminated. To make these two sets more consistent, we want to choose these sets from highly conserved regions between the genomes. Because as said before contaminations are caused by vectors that are failed to remove and sequenced into final genome assembly. If it is done with same vector multiple times in different genomes, then we can recover it by blast nearly perfectly and also this ensures the consistency of the contaminated data set. Now on when we use the word vector, it is not only vectors but also adapters, linkers, plasmids etc.

Aligning genomes of all metazoans against all is time expensive. So we chose genomes from species containing more vector content. To know which species contains more vector content, alignment of vectors against all genomes is done. Out of all alignment results, top 8 species with highest vector content have been chosen. The following steps from aligning genomes to create a model are shown in Fig.2. Now having 8 species at hand, aligning all 8 against 8 is about 56 whole genome alignments, which is also somehow time expensive. So to make it simple, we chose first 2 species and aligned them against other next 6 species and vice versa, which count to 26 alignments. For alignment we used 'megablast' with an E-value '1e-5' and identity at least 90%. Then each blast hit is a highly conserved region between the two selected genomes. In the next step, we took each conserved region and aligned it with all vectors. Here we also used megablast with same parameters as before. The resulting hits are conserved regions between the genomes with vector content. These are the sub regions on the genomes initially which we considered as contaminated sequences set. But after a manual observation of these sequences, we found that few of them are found to be in the protein coding genes which we are not sure to claim as contaminations. The reason behind this can be, while experimenting people attach CDS to vectors for cloning purposes. In vector databases, sometimes these vectors with CDS are listed. So to make this set more consistent, we aligned the sequences of the set against the 'exons'. All these exons are extracted from UCSC table browser (7) and are from RefSeq or Ensembl or Genscan databases. Sequences which are not at all aligned with the exons are finally considered as contaminated sequences. Valid non-contaminated sequences were randomly chosen from the highly conserved regions which lack any blast hits with vectors. To cluster similar sequences in both sets, we used 'blastclust' program with default options which are length coverage of 0.9 and score coverage threshold 1.75.

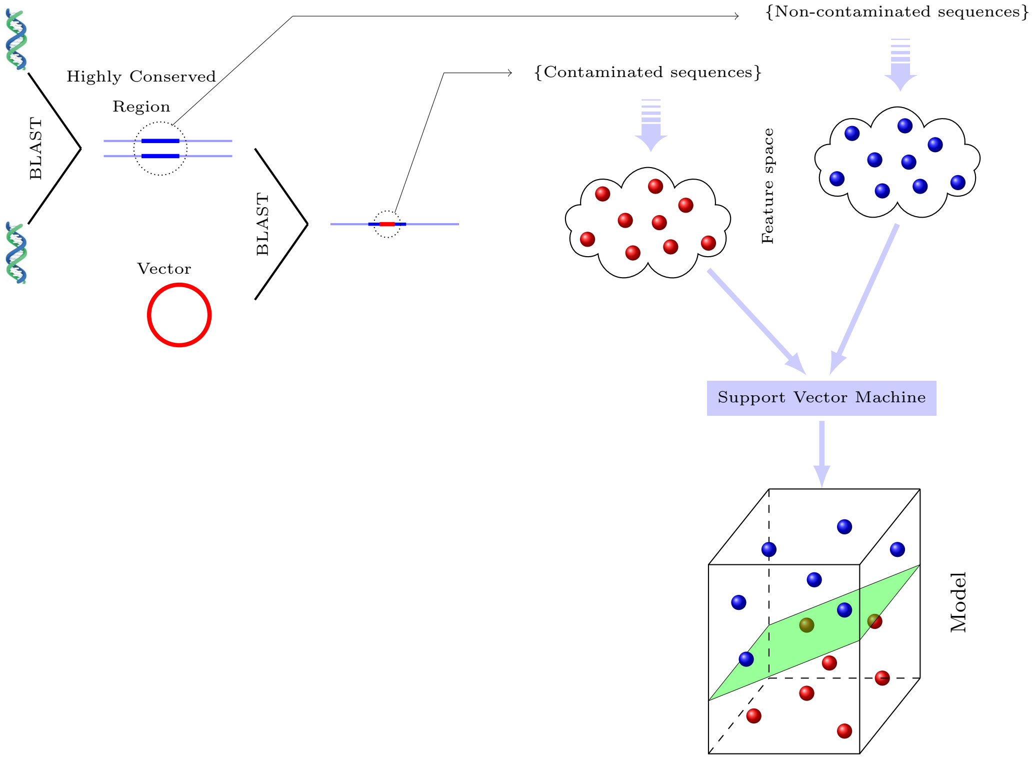

Fig. 2. Generation of a model that classifies contaminated and non-contaminated sequences

Having sets of contaminated and valid non-contaminated sequences, we had to find features that distinguish both sets. We worked on th e following features:

- k-mer count: number of k-mers per sequence
- k-mer distance: average distance between each of all k-mers in the sequence
- k-mer existence: existence of a k-mer as a binary value. If a k-mer exists, then value is 1 otherwise 0.
- k-mer mismatch scores: split the sequence into windows of size k. Then each k-mer and for each window count number of matched nucleotides. If all nucleotides matches score it to k. If there is one mismatch score to k-1 and so on. For each k-mer take the average of all such scores.
- GC content
- Stop codons: TTA, TAG, TGA
- Start codons: ATG, CTG, GTG, TTG

Here k is 1-3. After generating features, we had to train support vector machine and create a model out of features. Taking 80% of whole data sets from both sets, treating contaminated as positive and non-contaminated as negative data, we trained a model using Weka. We used Weka 3.7.3 (8) Sequential Minimal Optimization (SMO) scheme, polynomial kernel of degree 4 for this purpose. After tweaking between several options we got good model with these options. Here degree indicates, model with all combinations of 4 features worked together better that others. To check feasibility of the model, we tested our model on other sequence sets which were randomly chosen from ncRNAs, mRNAs, Rfam sequences and Human Watson strand UCRs with other species.

Results

At very first glance at the blast hits of vectors against genomes gave a nice prospect for vector contamination. Fig. 3 shows top 15 species containing at least 2000 hits per species. some of the hits On the top are *Gallus gallus* and *Danio rerio* which have 1072618 and 272003 hits respectively. Interestingly, we found *Homo Sapiens* with 75347 hits. From a well sequenced species one cannot expect these many hits. Most of the hits are actually subsequences of protein coding genes. In order to select contaminated species, blasting all blast hits (vectors against genomes) against exons has done.

All the steps done so far like blasting genomes against vectors in turn blasting them against exons is done only for selecting the genomes to work with. After all these steps, *Danio rerio* and *Gallus gallus* are prime candidates

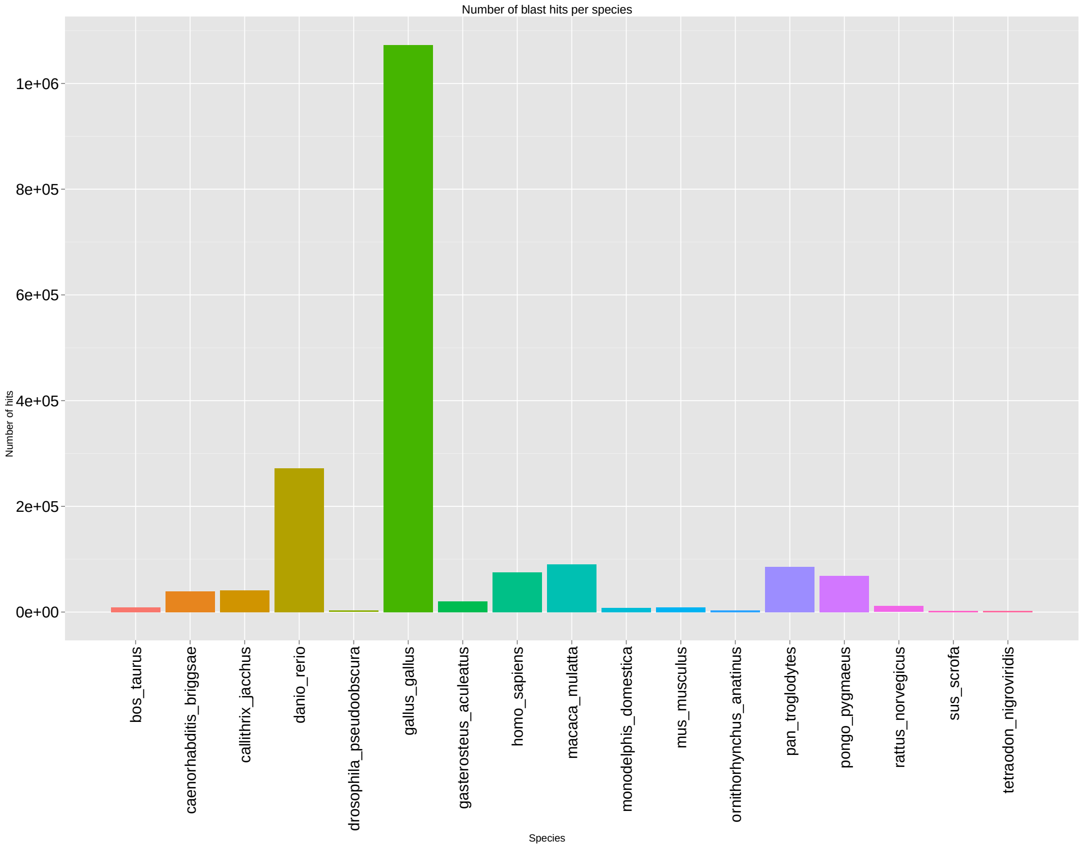

Fig. 3. BLAST hits vectors against genomes

to search for contaminations. So we chose *Gallus gallus* and *Danio rerio* and blasted against next top 6 species which are *Callithrix jacchus, Caenorhabditis briggsae, Gasterosteus aculeatus, Macaca mulatta, Mus musculus* and *Pan troglodytes* and vice versa. This yielded, 2852087 unique blast hits which mapped to 642885 highly conserved regions on *Danio rerio* and *Gallus gallus* together.

Blasting all these highly conserved regions against vectors lead to 10012 unique sequences. After excluding protein coding sequences (by BLAST against exons) and clustering we got 786 sequences which are in the contaminated set. For non-contaminated 1000 sequences were randomly selected from highly conserved regions which are not present in the blast hits against vectors. Fig. 4 shows different steps and the intermediate results discussed so far. Some stats on both sets of sequences are shown in Table 1.

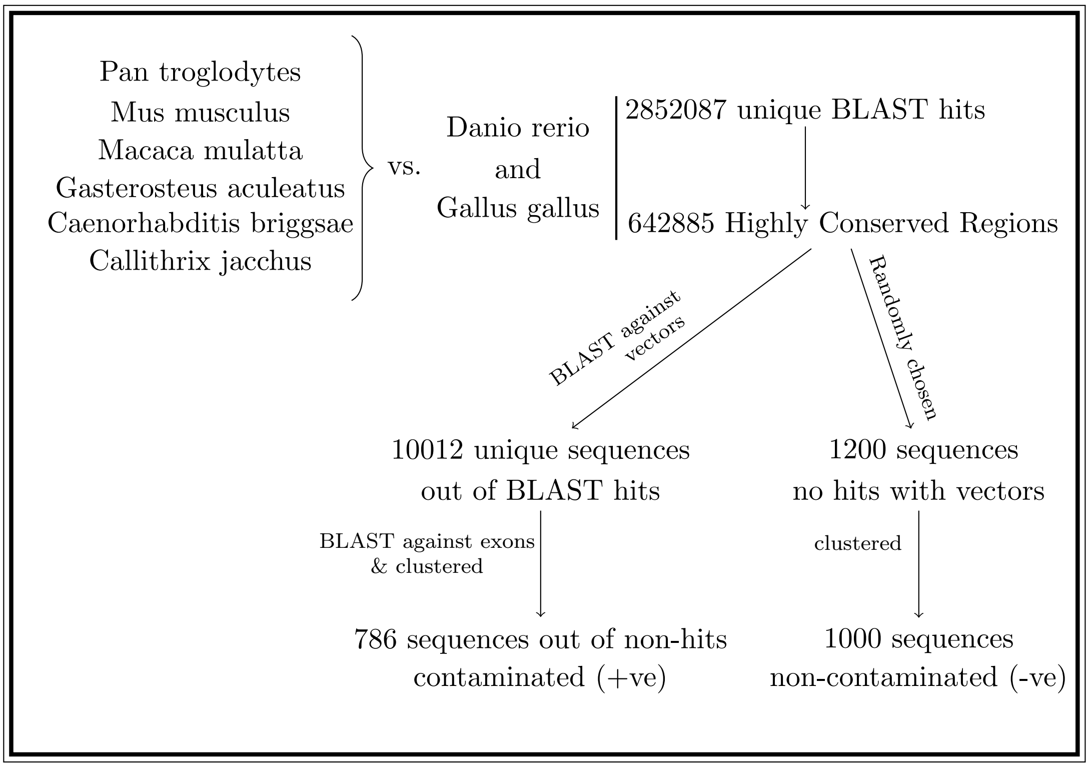
Pan troglodytes
Mus musculus
Macaca mulatta
Gasterosteus aculeatus
Caenorhabditis briggsae
Callithrix jacchus
vs.
Danio rerio
and
Gallus gallus
2852087 unique BLAST hits
642885 Highly Conserved Regions
BLAST against vectors
Randomly chosen
10012 unique sequences out of BLAST hits
1200 sequences no hits with vectors
BLAST against exons & clustered
clustered
786 sequences out of non-hits contaminated (+ve)
1000 sequences non-contaminated (-ve)

Fig. 4. Data sets extraction

Polynomial kernel of degree 4 with which the model is created says that there are several dependencies among the features. Table 2 lists the results of the classification on different test sets. Classification results also showed that our approach is promising. With 97.48% correctly and only 2.52% incorrectly classified instances, the initial cross validation demonstrated convincing classification performance of our model. For all other test sets the percentage of correctly classified instance are at least 91%. For human mRNAs 978 out

	Contaminated	Non-contaminated
Min. length	33	33
Max. length	528	683
Avg. length	91.01	105.11
Median	69.00	77.00
Standard deviation	55.07	86.07

Table 1
Stats of the contaminated and non-contaminated data sets

of 1000 sequences are classified as non-contaminated making a small error. 97.47% of human ncRNAs and 96.49% of watson strand human UCRs classified correctly. For ncRNAs the result varies for different species. For *Danio rerio, Mus musculus* and emphGallus gallus results are very similar. Remember our model is actually made up of conserved regions from emphDanio rerio and *Gallus gallus*. Still, from the results you can see our model classified better on human mRNA and ncRNAs than those of chicken and zebrafish. So, classification is in general not effected by model data,which is a requisite for a good model. But there are some classification errors, which can be very easy to see in case of cat ncRNA and Rfam sequences.

Test set	**Total Number of Instances**	**Correctly Classified Instances**	**Incorrectly Classified Instances**
Test set from existing data	357 Positives: 153 Negatives: 204	348(97.48%) True positives: 146 True negatives: 202	9(2.52%) False negatives: 7 False positives: 2
Danio rerio ncRNA	4431	4163 (93.96%)	268 (6.04%)
Felis catus ncRNA	738	677 (91.73%)	61 (8.27%)
Gallus gallus ncRNA	1102	1028 (93.29%)	74 (6.71%)
Homo sapiens ncRNA	2647	2580 (97.47%)	67 (2.53%)
Homo sapiens mRNA	1000	978 (97.80%)	22 (2.20%)
Mus musculus ncRNA	752	708 (94.15%)	44 (5.85%)
Rattus norvegicus ncRNA	757	704 (92.99%)	53 (7.01%)
Rfam sequences	786	718 (91.34%)	68 (8.66%)
Human UCRs Watson strand	2081	2008 (96.49%)	73 (3.51%)

Table 2
Classification results for different test sets

All ncRNAs from Ensembl ftp – http://www.ensembl.org/info/data/ftp/index.html

Human mRNA extracted from UCSC table browser – http://genome.ucsc.edu/cgi-bin/hgTables

Human Watson strand UCRs by Bejerano G – http://users.soe.ucsc.edu/ jill/ultra.html

Discussion

Contaminations are hazardous when working with sequences that are biologically significant. Care must be taken while analyzing such sequences, otherwise doubtful conclusions would be made. But primarily even more care should be taken while sequencing genomes. As sequencing the cleanest and precise sequences is hard to achieve, a tool that is made up of our strategy can be very advantageous to find the contaminations before working on some data. To know what went wrong with incorrect predictions, investigating the annotations on genes might also help.

References

[1] G. A. Seluja, A. D. Farmer, M. McLeod, C. Harger, P. A. Schad, Establishing a method of vector contamination identification in database sequences., Bioinformatics (1999) 106–110.

[2] G. Seluja, A. Farmer, M. McLeod, C. Harger, P. Schad, Establishing a method of vector contamination identification in database sequences, Bioinformatics 15 (1999) 106–10.

[3] J. Coker, E. Davies, Identifying adaptor contamination when mining DNA sequence data, Biotechniques 37 (2004) 194, 196, 198.

[4] B. L. Rhead, D. Karolchik, R. M. Kuhn, A. S. Hinrichs, A. S. Zweig, P. A. Fujita, M. Diekhans, K. E. Smith, K. R. Rosenbloom, B. J. Raney, A. Pohl, M. Pheasant, L. R. Meyer, K. Learned, F. Hsu, J. Hillman-Jackson, R. A. Harte, B. Giardine, T. R. Dreszer, H. Clawson, G. P. Barber, D. Haussler, W. J. Kent, The ucsc genome browser database: update 2010., Nucleic Acids Research (2010) 613–619.

[5] The UniVec database, `ftp://ftp.ncbi.nih.gov/pub/UniVec/`, accessed: 22/11/2010.

[6] S. Misener, VectorDB, `http://genome-www.stanford.edu/vectordb/`, accessed: 22/11/2010.

[7] D. Karolchik, A. S. Hinrichs, T. S. Furey, K. M. Roskin, C. W. Sugnet, D. Haussler, W. J. Kent, The ucsc table browser data retrieval tool., Nucleic Acids Research (2004) 493–496.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, SIGKDD Explor. Newsl. 11 (2009) 10–18.