A Generalized Pipeline For Cataloging Retrogenes

Ioana Bercea*
Department of Computer Science
University of Chicago

Shilpa Nadimpalli*
Department of Computer Science
Tufts University

Liqing Zhang Lenwood Heath
Department of Computer Science
Virginia Polytechnic Institute and State University

Abstract

The study of retrogenes is not only a relevant topic in the evolutionary analysis of gene origin, but also a way of performing genome-wide studies on possible correlations between different species. Existing literature on retrogene detection however, takes advantage of detailed annotations of certain genomes, therefore restricting their domain of application. This is what motivated the development of a generalized pipeline for retrogene detection. We detail the steps and main concepts in building this robust pipeline that, given the cDNA library and the genome database for a specific species, identifies retrogenes. Apart from being built genome-independent, our pipeline can also identify chimeric retrogenes. We also present possible improvements in light of the difficulties encountered in developing such a pipeline.

1 Introduction

1.1 Biological Background

Retrotransposons are genetic elements that are able to replicate and insert themselves into random locations on the chromosome. They are part of the bigger class of transposons. The "retro-" component in their name emphasizes the fact that, as opposed to DNA transposable elements who manifest themselves directly as DNA sequences, retrotransposons result out of an additional RNA - based intermediary step. With the help of reverse transcriptase, RNA copies of the original DNA sequence are translated back into DNA and then inserted into the genome. Structural characteristics of retrocopies(the retrontransposed copies) include the lack of introns and other regulatory genes, the presence of poly(A)-tails and flanking direct repeats.[**?**]

The fate of retrocopies depends on the location on the genome where they are transposed. We can distinguish between two cases: if transposed into an existing gene or not. In the first case, the retrocopy can be inserted into an intron and be transcriptionally silent,

*Research supported by CRA-W and CDC as part of the summer 2009 DREU program.

accumulating mutations with time. It can also disrupt the exon-intron structure, by becoming an exon. If transposed into an exon, it causes mutations and contributes to the cDNA transcript(such a gene would be called a chimeric gene). In all other cases, the retrocopy can decay and not ever be transcribed. If that happens, it is called a retro-pseudo-gene(as opposed to a retrogene). Sometimes, though, these retrocopies can recruit promoters and actually become functional.[**?**] Such genes are generally believed to be intron-less, though there is recent evidence that such retrogenes can gain introns.[**?**]

In terms of its importance, retrotransposition is a process through which genome size is increased and new genes are formed. Some of the genes affected by such mutations can even exhibit different new functions. Therefore, retrotransposition is an important element in the evolution of genomes. For example, studies in mammals and fruitflies have linked sex chromosome evolution to an increased expression of retrogenes.s[**?**] Retrogenes contributed to the formation of recent genes but also, influenced by natural/sexual selection, they were progressively recruited to enhance male germline function.[**?**] Other studies linked a retrogene to a human recessive disorder, gelatinous drop-like corneal dystrophy, a form of blindness.[**?**] Retrotransposition also allows us to trace the evolution and motions of genes[**?**] and do wide-scale and cross-species analysis of potential patterns with meaningful insights.[**?**]

1.2 The Pipeline

Computationally speaking, there does not exist a genome- independent methodology for indentifying retrogenes. Most procedures rely heavily on existing annotations, therefore introducing another possible source of error in their analysis. Existing literature is also divided into either identifying standard retrogenes(that evolve individually as genes) and indentifying chimeric genes(which would involve analysis of multiple genes at a time). Our pipeline tries to address both of these cases. The major inspiration for this methodology was a study done on the rice genome[**?**], but significant extensions have been added. The purpose of this paper is to detail this pipeline but also discuss the challenges and possible improvements in its development.

2 Methods

Our pipeline comprises several steps, each of them explained in the following subsections. First, we do a homology-based search through the target genome, using corresponding cDNA transcripts. Afterwards, we split our investigation into two major branches: looking for standalone retrogenes and looking for chimeric retrogenes. The difference is that the latter branch looks at exon-intron boundaries and considers multiple potential genes at a time. In contrast, the former branch looks at BLAST hits with a coarse granurality and is only interested in distinguishing between multi- and single-exon genes(as said before, retrogenes are expected to be single-exon). Once detected, potential retrogenes can undergo a series of other tests such as calculating frameshift, looking for Poly(A)- tails and calculating length of flanking regions used in age estimation. The major components in our pipeline are the algorithms involved in processing the BLAST results such that we can use them in relevant ways. Specifically, the Gluing Algorithm is used to assemble full-

length candidate genes from their BLAST fragments. In addition, the Grouping Algorithm was designed to aid us in detecting the groups of potential genes that might be involved in chimeric interaction. While both of the algorithms performed satisfactory, in the "Future Challenges" section, we also propose thoughts on enhancing them and adding extra functionality.

2.1 Initial Data Processing - BLAST

Our initial data consists of the target genome(downloaded from UCSC's Genome Browser[1]) and its corresponding cDNA sequences(downloaded from Ensembl Genome Browser[2]). The data was non-redundant and further pre-processing has been done in eliminating cDNA sequences shorter than 300bp, in accordance to previous methods.[**?**] The development of the pipeline has primarily been tested on the human and chimp genome, however, it is species - independent. The next step was to perform an all-to-all BLAST search.

Since our sequences where specified in nucleotides, we used the BLASTN algorithm to compare them. The cutoff e-value of 10^{-6} was chosen after some empirical observations guided by existing literature. [**?**][**?**] The task required extensive resources, so it was run through mpiBLAST[3] on a local VTech cluster of 20 nodes. Using the -m8 flag, we parsed our results into the following fields: percent identity, alignment length, number of mismatches, number of gap openings, e-value and bit score. The results were then uploaded into a MySQL database to facilitate further analysis.

2.2 Chaining BLAST hits - The Gluing Algorithm

The Gluing Algorithm was initially developed to optimize GeneWise input. Our initial goal was to assemble the BLAST hits into DNA sequences that would capture the length of a potential gene. (in order to avoid redundant results, we would expect to input into GeneWise possible genes rather than possible exons) Our main constraints were to respect the location on the chromosome of the hits, to take into consideration a reasonable gene structure and to avoid tandem duplication(to be detailed later). We also tried to assemble the individual characteristics that BLAST hits have, such as percent identity,alignment length and bit score. However, on the course of our project, we enhanced our algorithm to capture more relevant information such as an estimated number of exons. This component allowed us to distinguish between multi- and single- exon genes and can be further exploited for more information on the quality of the overall BLAST hit. We will further detail each of these components in the following sections.

I. Assembling the BLAST hits into potential genes

We will refer to BLAST hits as being represented by two main components: the portion of the cDNA query that matched (query_{start}, query_{end}) and the corresponding portion on the chromosome (subject_{start}, subject_{end}). The potential gene is represented as a portion on the chromosome (glue_{start}, glue_{end}) . When "gluing" BLAST hits, we also keep track

[1] genome.ucsc.edu
[2] www.ensembl.org
[3] www.mpiblast.org

of the last BLAST query we added to the glued portion. The necessity of these notations will become obvious in the following paragraphs.

In the preprocessing state, we order our BLAST hits according to subject$_{start}$. One also needs to observe that portions of the query can match on the chromosome either in the forward direction or in the reverse one, in which case we would have that subject$_{start}$ > subject$_{end}$. We distinguish between the two possible directions by adding another parameter to our database which has value 1 if direction is forward and -1 if direction is reverse. From this perspective, one assumption we made is that a potential gene will contain either BLAST hits that all matched in the forward direction, or in the reverse direction. Because of our initial goal, the algorithm is designed to look for contiguous portions on the chromosome in which all BLAST hits matched in the same direction. That is, when considering consecutive BLAST hits(after ordering them) to be glued in the same portion, we stop gluing once the direction changes and start a fresh portion that includes the current BLAST hit which matched in the opposite direction. Obviously, we also stop gluing and start anew once the chromosome or the query change. Another major factor that influenced the algorithm is the desire to avoid tandem duplication.(copies of the same gene that are relatively close to each other on the chromosome) That is the reason why, every time we decide to glue or not, we look at the last BLAST hit that we glued and check to see if our current hit seems to have been "glued" before. If that happens, we stop gluing and start a new portion. We control this situation by allowing an overlap of at most 20% for them to be considered as being part of the same gene, and not part of duplicate genes close to each other on the chromosome.The other design decisions involve assumptions about the structure of a gene: making sure that we never glue two BLAST hits that are too far apart and cannot represent exons of the same gene, or gluing two BLAST hits that matched to overlapping areas on the chromosome.(curr_subject$_{start}$ < prev_subject$_{end}$) We avoided the first situation by putting an upper bound on the intron size of 40,000bp.[**?**] The only cases in which we decide to glue the current BLAST hit to the current potential gene are either when it looks like it would be part of an already existing exon or when the difference between the exons they would represent is small enough for it to be an intron. For a pseudo code that summarizes the algorithm, please consult the following page.

II. Calculating quality of the resulting potential gene

Once we decide whether to add the current BLAST hit to our potential gene, we need to update the parameters that represent quality of the match.(percent identity,alignment length and bit score) Percent identity is added to a total quantity which, once we have the complete potential gene(after we stop gluing), will be averaged over the total number of BLAST hits that compose the gene. Because bit score is already normalized, it maintains its additive property, so it just gets added to a total bit score of the match. Calculating alignment length, however, requires a more complex analysis. Our algorithm will first check to see if the BLAST hits we want to glue overlap on the chromosome. If they do, we assume that they are part of the same exon and glue them. However, we do this by not actually looking at how the corresponding matches on the query behave. While this design decision does not affect the end result, it influences our calculation of alignment length in that it requires us to maintain a global list of the portions of the query that

Algorithm 1 Gluing Algorithm - Assembling BLAST hits

{Every quantity mentioned here is computed from the current BLAST hit we are reading and the last BLAST hit that we glued in the current potential gene}

while $query\_list \neq \emptyset$ **do**
  if query, chromosome and direction don't change **then**
    if *chromosome_overlap* **then**
      glue them {they're part of the same exon}
    else if $chromosome\_distance < 40,000$ **then**
      {they might be separated by an intron}
      if $query\_overlap < 20\%$ **then**
        glue {they are not part of duplicate genes}
        if $chromosome\_distance > 30$ **then**
          $exon\_count++$ {the distance between them is big enough to be an intron}
        end if
      else
        don't glue and start anew
      end if
    else
      don't glue and start anew
    end if
  else
    don't glue and start anew
  end if
end while

BLAST hit	query$_{start}$	query$_{end}$	subject$_{start}$	subject$_{end}$
1	794	913	11,460,136	11,460,017
2	132	797	11,461,819	11,461,154
3	306	658	11,461,834	11,461,479
4	180	613	11,461,834	11,461,401
5	97	135	11,462,341	11,462,303
6	1	98	11,463,366	11,463,269

Table 1: BLAST results for ENST00000279575, chr12, 11,460,017-11,463,366, reverse strand

get matched to our potential gene. Only after the potential gene is complete, we can proceed to calculate alignment length by comparing the percentage of the cDNA query that matched to the gene. In order to illustrate this, we can look at Table 1. According to Ensembl, the transcript ENST00000279575 originates from a gene on chromosome 12 Homo Sapiens, location 11,460,017-11,463,366 reverse strand. Our algorithm correctly identifies the location. BLAST also identifies the original exons located at positions on the query 1-98(hit 6), 97-135(hit 5), 132-797(hit 2) and 794-913(hit 1). However, in building our potential gene, we have two extraneous hits.(3 and 4) More than that, their query component overlaps almost completely with other query hits. Yet, our algorithm still glued them in the same potential gene. That is because, in considering them in order, their chromosome components overlap so we consider them to be part of the same exon and therefore, part of the same pontential gene. That is what motivated us in maintaining a global, non-redundant list of what portions of the query matched to our potential gene and evaluating alignment length only once our gene is complete. Further implications of this design decision will be continued in the "Discussion" and "Future Challenges" sections.

III. Estimating the number of exons of the potential gene

As we have seen in Table1, the number of BLAST hits that compose a potential gene does not necessarily reflect how many exons that potential gene has. From our empirical observations, BLAST will correctly identify real exons. Apart from that, however, BLAST will also return us partial matches which a lot of the times prove to be biologically meaningless. (they do not reflect either alternative splicing, or duplication of any sort) Our algorithm intends to go around that by doing a rough exon count of the potential gene. The exon count is done once we decide to glue a new BLAST hit. As one can see in Algorithm 1, we only increase the exon count when we decide that the BLAST hits are separated by an intron. As we have mentioned before, if the BLAST hits overlap on the chromosome, we consider them to be part of the same exon. Otherwise, we put a lower bound on intron size: BLAST hits that are at more than 30bp away from each other on the chromosome are far enough to allow an intron between them. [**?**] Applying this type of deduction for the example transcript in Table 1 gives us the correct exon count of 4.

IV. Detecting Retrogenes

Detecting retrogenes relies heavily on analyzing the quality of our potential genes. To start with, we only considered potential multi-exon genes(as returned by the Gluing algorithm) where percent identity 85% and alignment length > 95%. When it came to potential single-exon genes, we allowed mutations and constrained our search to regions with bit score > 400 and alignment length > 85%. From this new set, all single- exon potential genes originating from cDNA transcripts which also matched to multi-exon potential genes, were marked as potential retrogenes. To differentiate further between retrogenes and retropseudogenes, we calculated frameshift mutations and looked for premature stop codons. Although not implemented, the detection of poly(A)- tails and flanking direct repeats would be a natural step once we have our retrogenes narrowed down.

2.3 Identifying Chimeric Retogenes - The Grouping Algorithm and GeneWise

Detecting chimeric gene is a special case of retrogene detection because it involves analyzing the behavior of multiple gene at exon-intron level. Chimeric genes result by retrotransposition into an exon of an already existing gene.[**?**] Therefore, when looking for chimeric genes, we must analyze two genes: the one who initially generated the copy and the one in whose exon the copy got inserted. However, in order to be able to perform such an analysis, we need to have access to the exon - intron structure of the latter. That is where the GeneWise[4] algorithm comes into play. The intention to use GeneWise and optimize the input has dictated most of our design decisions and motivated our development of the Grouping algorithm. We also discuss the performance of GeneWise which fell short in providing reliable results. We also address this issue in the "Further Challenges" section, in which we propose methods through which the Gluing algorithm could be further expanded to eliminate the necessity of using an algorithm like GeneWise. In the end, we provide the methods for interpreting GeneWise output and detecting, based on the information available, chimeric genes.

I. The Grouping algoritm

Once BLAST hits are assembled into potential genes, we can use the information gained to filter out potential chimeric genes. The first step in the Gropuing algorithm clusters potential genes that overlap on the chromosome. We ordered the potential genes according to their start position and considered pairwise overlaps bigger than 95%. In the end, we are left with groups of genes which overlap almost completely. In order for these groups to exhibit meaningful chimeric behavior, we would expect them to include a single-exon gene(the retrocopy coming from a multi-exon parental gene) that overlaps completely with a multi-exon one. Therefore, we discarded groups that contain only multi- or single- exon genes. We also trace the single-exon genes to make sure that the transcript they represent has also matched to a multi-exon gene somewhere else on the genome. (what we would expect the parental gene to be) The genes that do not have any multi-exon potential parent are discarded and previous steps are again repeated for the new groups.

[4]http://www.ebi.ac.uk/Tools/Wise2/index.html

II. Formatting the data

The potential genes that survived the filtering process are then adjusted to be input into GeneWise. The cDNA transcripts are translated into protein sequence by choosing the forward frame which yield the longest open reading frame(ORF). Because of the existence of incomplete codons, the ORF length was calculated as being the distance between the first stop codon and the last one. Random scrutiny of the results lead us to believe that this was a viable approach in detecting the correct protein sequence. We then paired these cDNA transcripts with the regions of the chromosome to which they had matched.(the output we got from applying the gluing algorithm to our BLAST hits) These regions on the chromosome were flanked by 600 additional bps on each side. The recommended length of these flanking regions is 15000bp, however, such an amount would have significantly increased the running time. Such a big flanking region is also not necessary because our input was already resembling potential genes.(that was our initial motivation for developing the Gluing algorithm)

III. Interpreting GeneWise results

The translated cDNA transcripts and corresponding regions of the chromosome are then input into GeneWise. We recommend using a cluster of computers for this task, as the running time of GeneWise is significant. The output we obtained contained the locations of the exons(on the chromosome) and their corresponding bit scores. We collect all the non-redundant exons obtained for a specific region, in order to avoid alternative splicing. After we obtain the predicted gene structure, we go back to our groups and select the single-exon genes that matched one of the exons in a proportion $> 80\%$. These genes are marked as potentially chimeric and all the other ones are considered to have inserted themselves into an intron. Further calculations are made in order to detect premature stop codons and frameshift mutations when compared to the original transcript.

3 Results

We tested the pipeline by running it on the human and chimp genomes which, because of their extensive annotation, allowed us to cross-reference our results and fine tune our algorithms. Our Gluing algorithm is an essential component of the pipeline. By taking random results and comparing them to Ensembl annotations, we can say that the Gluing algorithm performs satisfactory in both assembling genes and analysis of the quality of the match. Exon counts, where available, were also exact. The Grouping algorithm is relatively fail-safe since it does not make any assumptions about the structure of genes. At each such steps, our search space narrowed down. For example, the Gluing algorithm allowed us to work with 662,911 potential genes for the human genome(as opposed to 2,156,725 BLAST hits) and 329,336 for the chimp.(1,232,690 BLAST hits) The grouping algorithm also allowed us to cut down our GeneWise input such that we obtained 39,080 cDNA queries(as opposed to 54,617 initially) and 595,160 potential genes for the human and 23,162 cDNA queries(as opposed to 33,032 initially) and 291,386 potential genes for the chimp. In the end, we identified 8,069 retrogenes and 8,070 retropseudogenes in the

human genome. Respectively, we found 1,859 retrogenes and 2,494 retropseudogenes in the chimp genome. Although GeneWise proved to be a major source of error in detecting chimeric genes, we found 688 retrocopies inserted into an existing exon (chimeric behavior) and 202 retrocopies transposed into an intron. For the chimp genome, the corresponding numbers were 534 for the chimeric case and 130 intronic retropseudogenes.

4 Discussion

For the human genome, our results were approximately equal to some in recent literature, [**?**],[**?**],[**?**], and higher than the ones in some older papers.[**?**],[**?**],[**?**] Differences in results could be explain by considering different cutoff values, but also by the accuracy of the data used. For example, one would expect that, since the chimp and human genome have roughly the same size, they would exhibit similar amounts of retrogenes and retropseudogenes. However, as one could observe from the "Results" section, there are fewer retrogenes in the human genome than in the chimp one. Also, the ratio between retrogenes and retropseudogenes is higher in the human genome than in the chimp genome. Such discrepancies can be explained by the difference in quality of the data available for the human genome versus the chimp genome. However, we designed this pipeline specifically to not depend on the quality of annotations and be able to perform a purely computational retrogene detection. Apart from depending on the performance level of the algorithms we use, the only other major source of error is the completeness of our initial data. Inaccuracies in sequencing the cDNA transcripts as well as the genome might lead to retrogenes not meeting the cutoff, as well as introducing false positives in our results.

Another major source of error in our pipeline is the use of GeneWise. As reported by the literature[**?**], GeneWise has been used by Ensembl in the final stages of gene structure prediction. However, even though we applied GeneWise when we had a rough estimate of a potential gene, the results were still confusing. Often times, our Gluing algorithm would correctly detect the gene from which the transcript originates(according to Ensembl), while GeneWise would fail to detect even the basic matches that BLAST had given us initially for that area. For example, we encountered genes that had 20 exons(correctly identified by BLAST and assembled by our Gluing algorithm) out of which GeneWise only output one who, ironically, was situated in an intron area as confirmed by the Ensembl annotation. Another problem we experienced was the existence of inadequately short single-exon genes(2-3 bps long). In order to avoid that, we required that all predicted gene structures had a minimum exon length of 40bp.

5 Future challenges

In light of the problems encountered, there are certain natural steps that could be made so that the pipeline becomes more efficient. Such changes will be explained in the first subsection. For the last part, we will discuss further questions that explore the existence and behavior of retrogenes. It is necessary, however, to keep in mind that contemporary algorithms have considerable upper bounds on their performance and also that, as long as the input genome and cDNA transcripts are poorly sequenced, our pipeline will always fall

short of representing biological truth.

I. The Gluing algorithm - the next level

Although initially built to simplify GeneWise input, the Gluing algorithm developed with the pipeline to contribute more and more to our analysis and we believe it has the potential to outgrow the pipeline. Firstly, the algorithm can grow in complexity when it comes to assembling potential genes from BLAST hits and evolve as a stand-alone algorithm for detecting gene-wide similarity searches. Alternatively, established algorithms could be used to perform the task.[**?**] [**?**] Another natural extension would be to enable the Gluing algorithm to perform gene structure prediction, to the point in which it could, hypothetically, replace GeneWise in the task of gene annotation. Conversely, we could also use other available annotation software such as Sim4, Est2gen or est_genome.[**?**]

II. Retrogene analysis

Once retrogenes are obtained, further analysis can be done, such as tracing parental genes. An obvious method would be to use "homolog families".[**?**] Within all the potential genes that a certain cDNA transcript matched to, we could do a pairwise comparison between our potential retrogene and all other candidate genes. If the closest neighbor is a single-exon gene, we could discard our potential retrogene and consider it to be a duplication of the parent.(rather than a retro-copy) Otherwise, if the closest neighbor is a multi-exon gene, then it is labeled as a parent. Once could also consider extending the nearest neighbor method and considering more than just one potential parent, in case similarity scores are not a particularly good criterion for distinguishing between them. Detecting parental genes could also allow for genome-wide analysis of what kind of genes are more likely to produce retrogenes, or what is the average ratio of retrotransposition, quantities which might be evolutionarily significant.

Another analysis which could be made takes place at the retrogene level. By comparing the retrogene to its parent, we can estimate the amount of mutations accumulated. Retrocopy age is also an important measurement and it is usually done by calculating the K_s value.(synonymous mutations at each site)[**?**] [**?**] [**?**] One could also estimate age without considering the structure of the parental gene.(in case the parent cannot be traced or it does not exist any more) In such cases, one could measure the length of the poly-A tail or compare the two flanking regions of the retrogene.

III. Retrogene theory

On a more theoretical level, one could also design a probabilistic model for the birth and death of retrogenes. Once analysis of retrogenes is performed, such as age estimation and parent detection, one could ideally assign probabilities to each stage of the retrocopy survival process. Some of the natural questions that occurred during our research include the issue of time affecting the survival rate of retrogenes. Can we trace the life of retrogenes and detect certain critical moments?(for example, how many retrocopies survive immediately after they are transcribed? how many retrocopies from the same parent have the same age?) Other questions involve the locations in which these retrocopies are inserted.

How "random" is the place where these retrogenes are transcribed? Are there "preferred" places on the genome or can we find a correlation between the initial parental place and the retrocopy's place?

References

[1] Adel, Khelifi, Duret Laurent and Mouchiroud Dominique, "HOPPSIGEN: A Database of Human and Mouse Processed Pseudogenes", *Nucleic Acids Research*, 33, D59-D66, (2005)

[2] Birney, Ewan, Michele Clamp and Richard Durbin, "GeneWise and GenomeWise", *Genome Research*, 14, 988-995, (2004)

[3] Buzdin, A. A., "Retroelements and formation of chimeric retrogenes", *Cellular and Molecular Life Sciences*, 61, 2046-2059 (2004)

[4] Deutsch, Michael and Manyuan Long, "Intron-Exon Structures of Eukaryiotic Model Organisms",*Nucleic Acids Reserach*, 27, 3219-3228, (1999)

[5] Drouin, Guy, "Processed Pseudogenes Are More Abundant in Human and Mouse X Chromosomes than in Autosomes",*Molecular Biology and Evolution*,23,1652-1655, (2006)

[6] Fablet, Marie, Manuel Bueno, Lukasz Potrzebowski and Henrik Kaessmann, "Evolutionary Origin and Functions of Retrogene Introns", *Molecular Biology and Evolution*, 26, 2147-2156(2009)

[7] Kaessmann,Henrik, Nicolas Vinckenbosch and Manyuan Long, "RNA-based gene duplication: mechanistic and evolutionary insights", *Nature Reviews Genetics*, 10, 19-31 (2009)

[8] Marques, Ana Claudia, Isabelle Dupanloup, Nicolas Vinckenbosch, Alexandre Reymond, Henrik Kaessmann, "Emergence of Young Human Genes after a Burst of Retroposition in Primates", *Public Library of Science Biology ONE* ,3, 1970-1979(2005)

[9] Miao, H.E., L.I. Jidong and Shanghong Zhang, "Statistical Characteristics of Eukaryiotic Intron Database", *Frontiers of Biology in China*, 4, 363-366, (2006)

[10] Pan, Deng, Liqing Zhang, "Burst of Young Retrogenes and Independent Retrogene Formation in Mammals", *Public Library of Science Biology ONE*, 4, (2009)

[11] Rohozinski, Jan, Dolores J. Lambbc, Colin E. Bishopad, "UTP14c Is a Recently Acquired Retrogene Associated with Spermatogenesis and Fertility in Man", *Biology of Reproduction*, 74,644-651 (2006)

[12] She, Rong, Jeffrey S.-C. Chu, Ke Wang, Jian Pei and Nansheng Chen, "genBlastA: Enabling BLAST to Identify Homologous Gene Sequences",*Genome Research*, 19, 143-149, (2009)

[13] Vinckenbosch,Nicolas, Isabelle Dupanloup and Henrik Kaessmann, "Evolutionary fate of retroposed gene copies in the human genome", *Proceedings of the National Academy of Science*, 103, 3220-3225, (2006)

[14] Wang, Wen, Hongkun Zhengb, Chuanzhu Fand, Jun Lib, Junjie Shib, Zhengqiu Caib, Guojie Zhanga, Dongyuan Liub, Jianguo Zhangb, Søren Vangg, Zhike Lub, Gane Ka-Shu Wongb, Manyuan Longd, and Jun Wang, "High Rate of Chimeric Gene Origination by Retroposition in Plant Genomes", *The Plant Cell*, 18,1791-1802 (2006)

[15] Zhang, Hongyu, "Alignment of BLAST High-scoring Segment Pairs Based on the Longest Increasing Subsequence Algorithm",*Bioinformatics*, 19, 1391-1396, (2003)

[16] Zhang, Zhaolei, Paul M. Harrison, Yin Lin and Mark Gerstein, "Millions of Year of Evolution Preserved: A Comprehensive Catalog of the Processed Pseudogenes in the Human Genome", *Genome Research*, 13, 2541-2558, (2003)