Unravel motifs in UTRs and introns

Andreas Jangmo[1,☯], Johan Lord[2,☯]

1,2 Royal Institute of Technology (KTH)/School of Computer Science and Communication (CSC), Stockholm, Sweden

☯These authors contributed equally to this work.

Abstract

A sequence logo is a graphical representation of conserved bases in a sequence of DNA or protein. It is similar to a bar graph with the bars being stacks of letters corresponding to the nucleotides or amino acids. The logo is created from a file of aligned sequences and the size of the letters correspond to the conservation of that base at the position in the sequence specified on the x-axis. The logo can be used to illustrate a specific motif or the presence of functional units or protein binding sites in DNA sequences. In this report we have studied what the sequence logos are for the the regions before and after the translation start site and the first intron of each gene in the human genome. In doing so we first extracted the sequences for the regions of interest using Ensamble's Biomart. Using these extracted sequences we aligned them using python and created sequence logos using the python package Biopython and Weblogo. We end with a brief discussion and an interpretation of the resulting logos.

Introduction

In genetics, a motif is a pattern in nucleotide or amino acid sequences that have, or are thought to have, a biological significance. Motifs are important as they may determine a protein's secondary structure when present in exons. It may also indicate binding sites for a large variety of proteins such as enzymes or more direct RNA level processes such as ribosome binding. Some proteins bind with very high specificity such as Type II restriction enzymes [1]. These are part of the immune system in bacteria where they destroy viruses by splicing them up. Deviating from their binding sites could thus lead to catastrophic results. More common is that motifs vary more in composition such as the TATA box that indicates the binding site for RNA polymerase. It is apparently very rare to find a promoter that matches this sequence exactly [1]. A convenient tool when analyzing motifs is therefore to construct something like a histogram or a bar graph illustrating the frequency of each nucleotide at each position in a region of DNA from a large number of aligned sequences. This is done by creating something called a sequence logo [2]. A sequence logo is a graphical way of representing the variation of nucleotides or amino acids around a certain site that makes it easier to find candidate motifs. Each position relative to the site is assigned a score of information content, that is essentially a measure of the distance from a state of randomness that per definition has no information. The information content is measured in bits from complete random being 0 to perfect conservation represented by 2 bits as there are $2^2 = 4$ types of nucleotides. The actual letters representing the four nucleotides in our case are then scaled with

their information content at each position in the sequence thus resulting a diagram called a sequence logo.

The purpose of our work has been to create sequence logos for regions in the human genome. The regions we have focused on are the ones before and after translation start site and before and after the beginning and end of the first intron of those genes that has at least one intron. Creating sequence logos for these regions might hopefully reveal conserved patterns that indicate the presence of motifs.

Materials and Methods

For the genetic information we need we have used Ensamble's Biomart [3] mostly because of our prior knowledge of the application but also for its popularity. As a programming language we choose Python and more specifically the package Biopython [4]. The python programs we have written and use throughout this project are mergeseq.py, cutseq.py and createlogo.py.

Gathering gene data

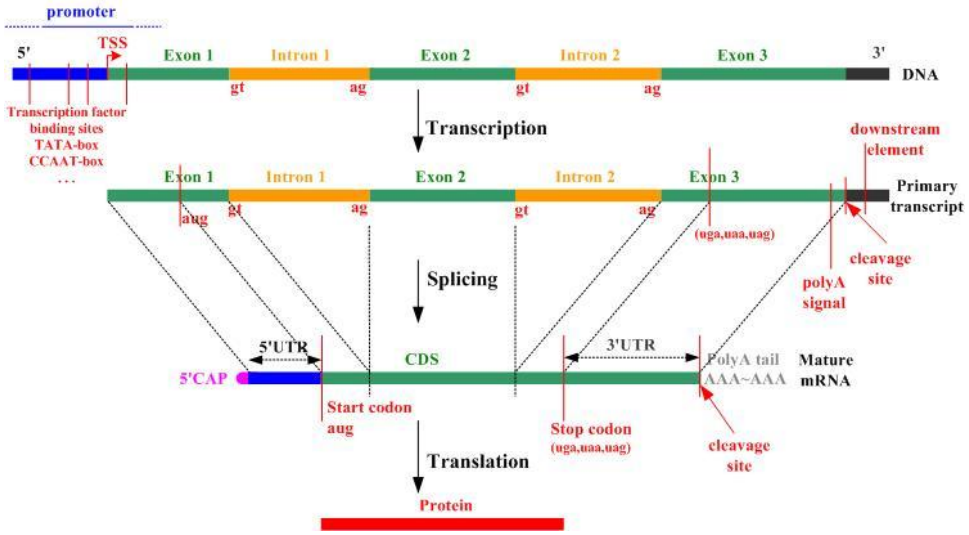

Figure 1. An illustration of a gene showing what happens to different regions through the processes of transcription, splicing and translation. [5]

Having decided on using as much of the human genome as possible we gathered the data using Biomart. We have used the december 2013 Homo sapiens high coverage assembly GRCh38 ("Ensambl Genes 83"/GRCh38.p5). We divided this work up into two parts: retrieving the sequences before and after the translation start site and retrieving the sequences from the beginning and end of the first intron.

Solving the first part was easy enough. From the dataset mentioned above (GRCh38.p5) we selected only coding sequences as these all start with the initiation codon. Further we added 15 nucleotides upstream to these sequences in our query to include an appropriate number of nucleotides before and after the start codon for our analysis. This was done on chromosomes 1-22, X and Y. An URL query to Biomart for this data, called Query 1, is attached as a reference [7].

In retrieving the introns this proved to be a bit more tedious as Biomart does not directly supply intron sequences. Studying the structure of a gene, seen in *Fig* 1, one

could though come to some conclusions. A gene include both non coding parts such as UTR:s and introns, and coding parts which consist of parts of one or more exons. Exons include the 5' and 3' UTR:s and a number of exons are spliced out to assemble a transcript which are further spliced to become a coding sequence. Thus taking the difference of the complete genes and the exons will leave the introns. As we also want to include regions upstream the start of the introns as well as downstream the end of them we achieve this by using nucleotide coordinates. This method is also convenient for other more obvious reasons as we are dealing with large amounts of data (all unspliced genes from the human genome accumulates to approx. 1.8 Gb of data). Picking out the sequences of interest by using coordinates limits the amount of data we would have to process and therefore greatly improves processing times.

We retrieved the whole unspliced genes for chromosomes 1-22, X, Y and the Exon sequences for the same using Query 2 [8] and Query 3 [9] respectively. The actual sequences of the exons might seem a bit redundant as we are only using the coordinates of the exons and as they are already present in the unspliced gene. However these were included for control purposes, to fail check our software and make sure it really did pick out the correct sequences as introns and exons.

Processing gene data

Using the data from the previous section we now continue by filtering out the sequences to use as a base for the logo. We then need to align them in a text file in preparation for the logo creation. We implement all this using Python and the two programs doing all the work for this part are mergeseq.py and cutseq.py.

As the extraction of coding sequences of already includes 15 nucleotides upstream the only task nescessary to retrieve motifs around the coding start site was to cut each sequence at a desired length. These operations are performed by *cutseq*. Exons are only provided as they appear in transcripts, thus they do not individually define exon regions in the unspliced gene. The program *mergeseq* solves this by using exon coordinates for each transcript and the gene strand on which it resides. Coordinates are provided as base pair start and end for the gene or exon which ignores that sequences of genes on the negative (antisense) strand are still provided in the $5' \rightarrow 3'$ direction. For a gene, G^+, on the positive (sense) strand the start, s, and end, e, position of a subsequence, g^+, relative to the chromosome coordinates G_a^+ and G_b^+ will be: $g_s^+ = g_a^+ - G_a^+ + 1$ and $g_e^+ = g_b^+ - G_a^+ + 1$. For a gene on the negative strand, G^-, we get start $g_s^- = G_b^- - g_b^- + 1$ and end $g_e^- = G_b^- - g_a^- + 1$. An example is given in 2.

base pair	5	6	7	8	9	10	11	12	
5'	A	**C**	**G**	**A**	T	C	C	A	3'
3'	T	G	C	T	**A**	**G**	**G**	T	5'

Figure 2. As can be seen the example sequence above would have has $G_a^{+,-} = 5$ and $G_b^{+,-} = 12$ while the subsequence (in bold) on the positive strand goes **C-G-A** and has the relative start $g_s^+ = 6 - 5 + 1 = 2$ and end $g_e^+ = 8 - 5 + 1 = 4$. The subsequence on the negative strand reads **G-G-A** with start $g_s^- = 12 - 11 + 1 = 2$ and end $g_e^- = 12 - 9 + 1 = 4$.

The nescessary task to retrieve the first intron is thus to merge all overlapping exon coordinates and create a set of all disjunct regions. For a gene G with n of these exon regions $e_{g,i,a}$,$e_{g,i,b}$ indexed i as they appear in the unspliced $5' \rightarrow 3'$ there are $n-2$ intron regions (as each gene ends with an exon) where the first intron has the start and end position $e_{g,1,b+1}$, $e_{g,2,a-1}$ respectively.

Creating sequence logos

For sequence logo creation there are a few tools at our disposal. We choose to use Steven Brenner's WebLogo [6] for its simplicity and the possibility of accessing this tool via the function weblogo() in the Biopython class motifs. Weblogo is mainly used through their web application which we initially used for experimental purposes. It can also be used by downloading their source code, by using their python package or from a third party software such as Biopython. We choose to implement this using the above mentioned function weblogo() in Biopython. Weblogo uses a multiple sequence alignment as a basis for logo creation. Three file formats for these alignments are available: FASTA, ClustalW and Flat. Flat format means that the sequences are just listed on top of each other without sequence names or other header information. Since we had no use for header information at this point, flat format was definitely most suited for our purposes. The simplicity of the flat format also made scripting a bit simpler.

Using the flat formatted alignment files created in the previous step a python program called createlogo.py goes through each line of the file and adds that sequence as a Biopython Seq object to a list. From this list a Biopython motif object is created and from this object the logo is created by connecting to the weblogo service via the function weblogo() in the motifs class. Thus one needs an internet connection to use this program. The output of createlogo.py is simply a PNG file displaying the logo.

Results and Discussion

The three resulting logos from the region around the start codon and around the beginning and end of the first intron can be seen in *Fig* 3. For the logo of the region around the start codon there did not seem to be any easy way to ignore the start codon (as this obviously blows up) so we choose to show it as it is but adjusted the y axis to show more information of the other areas.

As we can see in 3a, GCC and ACC seems higly conserved right before the start and G seems to be conserved directly after it. This is quite expected as it agrees with a well known consensus sequence found in eukaryotic mRNA known as the Kozak sequence [10], after Marilyn Kozak. This has the consensus (gcc)gccRccAUGG where R stands for a purine (A or G). The Kozak sequence is not the same as the ribosomal binding site but is recognized by the ribosome and plays a huge role in the initiation of translation.

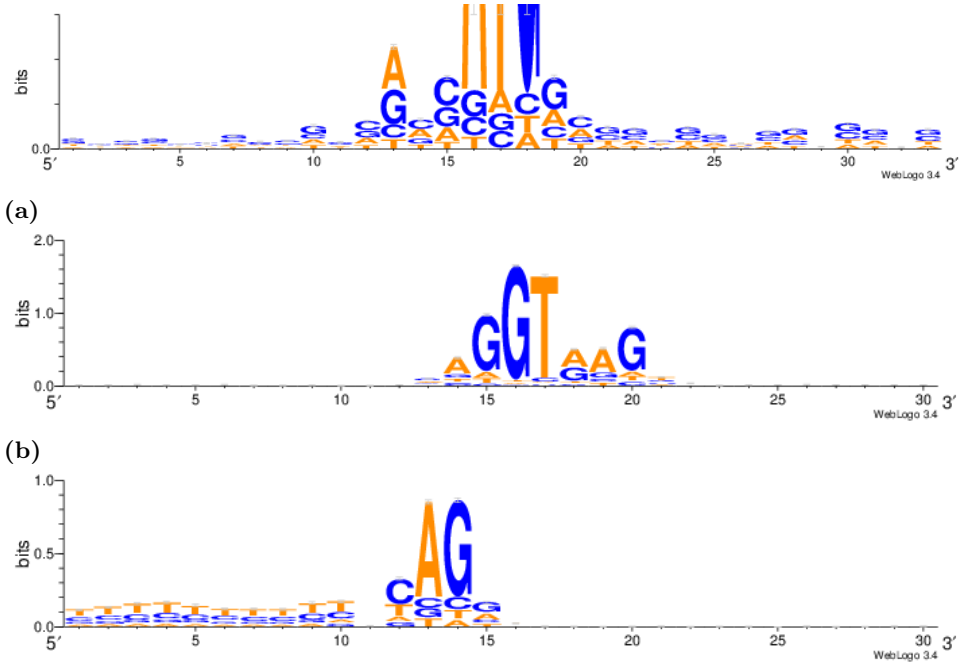

Figure 3. (a) The logo for the sequence 15 nucleotides upstream to 15 nucleotides downstream the initiation codon. (b) The logo for the sequence 15 nucleotides upstream to 15 nucleotides downstream the start of the first intron. (c) The logo for the sequence 15 nucleotides upstream to 15 nucleotides downstream the end of the first intron.

References

1. D'haeseleer Patrik, What are DNA sequence motifs? Nature Biotechnology 24, 423 - 425 (2006)

2. Schneider, T.D. & Stephens, R.M. Sequence Logos: a new way to display consensus sequences. Nucleic Acids Res. 18, 6097–6100 (1990).

3. Ensembl release 83, December 2015 © WTSI / EMBL-EBI http://www.ensembl.org/index.html

4. Biopython 1.66, released on 21 October 2015. http://biopython.org/wiki/Main_Page

5. Prof B. Jayaram & Co-workers, Genome Tutorials. Supercomputing Facility for Bioinformatics & Computational Biology, IIT Dehli. As of 2016-01-03 avaible here: http://www.scfbio-iitd.res.in/research/genomics.html

6. Crooks GE, Hon G, Chandonia JM, Brenner SE WebLogo: A sequence logo generator, Genome Research, 14:1188-1190, (2004)

7. Query 1: Coding sequences for chromosome 1-22, X, Y with 15 nucleotides upstream flank, Header: Gene ID, Ensembl release 83, December 2015 © WTSI / EMBL-EBI

   `http://www.ensembl.org/biomart/martview/e8279e04f1de2a2309c53c23a7a5aa05?VIRTUALSCHEMANAME=default&`

ATTRIBUTES=hsapiens_gene_ensembl.default.sequences.ensembl_gene_id|hsapiens_gene_ensembl.default.sequences.coding|hsapiens_gene_ensembl.default.sequences.upstream_flank."15"&FILTERS=hsapiens_gene_ensembl.default.filters.chromosome_name."1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,X,Y"&VISIBLEPANEL=resultspanel

8. Query 2: Unspliced gene for chromosome 1-22, X, Y, Headers: Gene ID, gene start, gene end, Ensembl release 83, December 2015 © WTSI / EMBL-EBI

   http://www.ensembl.org/biomart/martview/e8279e04f1de2a2309c53c23a7a5aa05?VIRTUALSCHEMANAME=default&ATTRIBUTES=hsapiens_gene_ensembl.default.sequences.ensembl_gene_id|hsapiens_gene_ensembl.default.sequences.start_position|hsapiens_gene_ensembl.default.sequences.end_position|hsapiens_gene_ensembl.default.sequences.gene_exon_intron&FILTERS=hsapiens_gene_ensembl.default.filters.chromosome_name."1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,X,Y"&VISIBLEPANEL=resultspanel

9. Query 3: Exon sequences for chromosome 1-22, X, Y, Headers: Gene ID, exon start, exon end, strand, Ensembl release 83, December 2015 © WTSI / EMBL-EBI

   http://www.ensembl.org/biomart/martview/e8279e04f1de2a2309c53c23a7a5aa05?VIRTUALSCHEMANAME=default&ATTRIBUTES=hsapiens_gene_ensembl.default.sequences.ensembl_gene_id|hsapiens_gene_ensembl.default.sequences.gene_exon|hsapiens_gene_ensembl.default.sequences.exon_chrom_start|hsapiens_gene_ensembl.default.sequences.exon_chrom_end|hsapiens_gene_ensembl.default.sequences.strand&FILTERS=hsapiens_gene_ensembl.default.filters.chromosome_name."1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,X,Y"&VISIBLEPANEL=resultspanel

10. Kozak, M (31 January 1986). "Point mutations define a sequence flanking the AUG initiator codon that modulates translation by eukaryotic ribosomes.". Cell 44 (2): 283–92