**Bioinformatics / Computational Biology** - **Group 10**

25/10/2019

LAB #3 – Probabilistic Models

# **Group i**

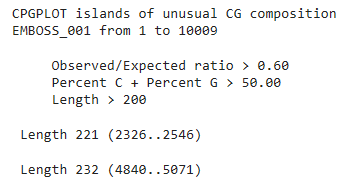
## **a.**

**CpGPlot**

This tool tries to identify CpG islands by using four parameters (window size – WS -, minimum length of an island – ML -, minimum ratio observed/expected of CpG to C + G - MR - and a minimum average percentage of C + G - MP) to analyze a given sequence in the described way: a window of size WS is slid along the sequence. For each position of the window, the ratio of CpG to C + G is calculated. The criterion to define a CpG island in a determined region of the sequence is verified if, over an average of 10 windows AND not less than ML bases, calculated (%G + %C) content is over MP AND the calculated Observed/Expected ratio is over MR.

Given this, it should be pointed out the way that the ratio Observed/Expected ratio is calculated: while the observed number of CpG patterns in a window is simply the number of times a 'C' is found followed immediately by a 'G', the expected number is number of CpG you would expect to see in that window based on the frequency of C's and G's in that same window. For the expected number, it is assumed that P(C) and P(G) are independent, leading to the following expression:

As an illustrative example, it was run the CpGPlot with the default parameters: WS=100, ML=200 , MR=0.6, MP=50. This tool allows not only a graphical perspective of the C and G content of the input sequence but also a brief summary of the CpG islands found, as represented below:

Uma imagem com texto

Descrição gerada automaticamente

In order to have some insight on the different parameters, some tests were carried out while varying parameters. Those results are exposed on the following table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Variable Tested* | *Results* | | | | *Observations* |
| **WS** | **ML** | **MR** | **MP** |
| *WS* | 20 | 200 | 0.6 | 50 | * With the increase of the window size, the graphical representations of the Obs/Exp and of the percentage of C+G become smoother, since for each iteration, more elements are considered. Thus, the variations from one position to the next one are smaller; * The number of CpG islands increases with the windows size. This is due to the fact that the smaller the window, the more sensitive is the criterion of CpG islands identification to non-CpG regions inside a CpG island (if, by chance, there’s a small region where the C+G content is smaller, it’ll have repercussions on this criterion verification directly); * It’s important that the regions considered to be a CpG island when WS = 100 are different from those when WS = 500. |
| No CpG islands | | | |
| 100 | 200 | 0.6 | 50 |
| 2 CpG islands:  Length 221 (2326..2546)  Length 232 (4840..5071) | | | |
| 500 | 200 | 0.6 | 50 |
| 6 CpG islands:  Length 356 (251..606)  Length 249 (636..884)  Length 822 (2151..2972)  Length 270 (3560..3829)  Length 384 (7573..7956)  Length 721 (8343..9063) | | | |
| *ML* | 100 | 50 | 0.6 | 50 | * There are no differences on the graphical representations of the Obs/Exp and of the percentage of C+G, since all the example are calculated using the same window size; * The number of CpG islands increases as the minimum length required for a region of the sequence to be considered a CpG island decreases. This fact is intuitive since we may have small length sequences that fit the C+G percentage and Obs/Exp ratio parameters but as we increase the sequence length they will be influenced by regions that don’t fit the parameters and so overall they won’t be considered CpG islands. In fact the only two regions that are present in both the 50 and 200 ML cases are the ones which in ML = 50 have more than 200 bases (and nor of them has more than 500 bases), as expected. |
| 18 CpG islands:  Length 123 (271..393)  Length 192 (504..695)  Length 95 (859..953)  Length 128 (2012..2139)  Length 221 (2326..2546)  Length 120 (2656..2775)  Length 92 (3695..3786)  Length 65 (4181..4245)  Length 232 (4840..5071)  Length 102 (5245..5346)  Length 51 (5562..5612)  Length 58 (5791..5848)  Length 51 (6038..6088)  Length 59 (6764..6822)  Length 181 (7584..7764)  Length 104 (8540..8643)  Length 118 (8781..8898)  Length 60 (9752..9811) | | | |
| 100 | 200 | 0.6 | 50 |
| 2 CpG islands:  Length 221 (2326..2546)  Length 232 (4840..5071) | | | |
| 100 | 500 | 0.6 | 50 |
| No CpG islands | | | |
| *MR* | 100 | 200 | 0.2 | 50 | * There are no differences on the graphical representations of the Obs/Exp and of the percentage of C+G, since all the example are calculated using the same window size; * As expected, as the minimum ratio Obs/Exp required increases, the number of CpG islands decreases, since there will be less regions verifying the CpG identification criterion. * An interesting point is that as the MR increases, the CpG islands for the new MR are always contained in one of the CpG islands for the previous MR. This happens due to the fact that as MR, the more restrictive the CpG island identification criterion becomes, and so some bases that are lowering the ratio are dropped out by the algorithm. |
| 4 CpG islands:  Length 221 (2326..2546)  Length 328 (4780..5107)  Length 238 (7527..7764)  Length 323 (8413..8735) | | | |
| 100 | 200 | 0.6 | 50 |
| 2 CpG islands:  Length 221 (2326..2546)  Length 232 (4840..5071) | | | |
| 100 | 200 | 0.8 | 50 |
| 1 CpG islands:  Length 218 (4848..5065) | | | |
| *MP* | 100 | 200 | 0.6 | 20 | * There are no differences on the graphical representations of the Obs/Exp and of the percentage of C+G, since all the example are calculated using the same window size. * As expected, as the minimum percentage of (%G + %C) content required increases, the number of regions considered CpG islands decreases, as the CpG island criterion becomes stricter. * Again, as happened with the MR analysis, the CpG islands for a higher MP are contained in the CpG islands considered for a smaller CpG, exactly by the same reason for the MR analysis. |
| 6 CpG islands:  Length 206 (271..476)  Length 450 (504..953)  Length 336 (1949..2284)  Length 486 (2290..2775)  Length 232 (4840..5071)  Length 340 (7584..7923) | | | |
| 100 | 200 | 0.6 | 50 |
| 2 CpG islands:  Length 221 (2326..2546)  Length 232 (4840..5071) | | | |
| 100 | 200 | 0.6 | 80 |
| No CpG islands | | | |

**CpG Islands**

This tool works in the same way of the CpGPlot, but always with WS = 200, MR = 0.6 and MP = 50. In fact, this method was firstly described by *Gardiner-Garder and Frommer (1987)* and, so, it is often used for this kind of tasks. One extra of this tool in comparison with CpG plot is that it presents the result for the Obs/Exp ratio and %GC in each window that verifies the CpG identification window.

The results for this tool work window-wisely, *i.e.*, it applies the CpG island identification criterion for each window and presents each window that verifies it as a CpG identification. Note, that this implementation implies that the minimum length required for a CpG island is 200 bp. Thus, given that this tool isn’t able to merge the different the different windows, it was post-processed by us, where we defined the extension of the CpG island from the first index of a given window to the last index of contiguous windows (*i.e.*, separated by 1bp). Those results are presented below, in Table:

|  |  |
| --- | --- |
| ***Bp Interval (bp)*** | ***Merged Sequences (bp)*** |
| 0 – 1k | 160 – 373; 469 – 673;  482 – 687; 495 – 723;  531 - 730 |
| 1k – 2k | - |
| 2k – 3k | 2315 – 2549; 2366 – 2569 |
| 3k – 4k | 3651 – 3871; 3674 – 3873;  3678 – 3878; 3685 – 3893;  3701 – 3900; 3705 – 3904; |
| 4k – 5k | 4745 – 4944; 4747 – 4954;  4766 – 4965; 4769 – 4976;  4785 – 5001; 4804 – 5003;  4806 – 5005; 4853 – 5080;  4883 – 5088 |
| 5k – 6k | 4926 – 5125; 4934 – 5137 |
| 6k – 7k | - |
| 7k – 8k | 7461 – 7660; 7482 – 7835;  7638 – 7837 |
| 8k – 9k | 8403 – 8602; 8414 – 8613;  8416 – 8627; 8457 – 8679;  8493 – 8704; 8509 – 8716;  8566 – 8765; 8569 – 8765;  8668 - 8872 |
| 9k – 10k | - |
| 10k – 10009 | - |

As it can be seen, it considers lots of overlapping sequences (even after the merging of contiguous windows) which might be all part of the same CpG island.

In order to confirm the results provided by CpG Islands, it was run the CpGPlot tool with the following inputs: WS =200; ML = 200; MR = 0.6; MP = 50. The results yielded by this tool were that there are 5 CpG island:

1. 1st sequence - Length 247 (489 - 735);
2. 2nd sequence - Length 293 (2287 – 2579);
3. 3rd sequence - Length 206 (4830 - 5035);
4. 4th sequence - Length 256 (7531 - 7786);
5. 5th sequence - Length 411 (8538 - 8948)

Thus, though the concordance between the results of both tools are not 100% concordant, there’s clearly a relationship between them, since we can correlate/identify (since they overlap partially) the following CpG islands:

|  |  |  |
| --- | --- | --- |
| ***From CpGPlot*** | ***From CpG Islands*** | ***Location (Approximately)*** |
| 1st sequence | 2nd to 5th merged sequence in 0 - 1k (bp) | 470 - 735 |
| 2nd sequence | All merged sequences in 2k - 3k (bp) | 2300 - 2550 |
| 3rd sequence | All merged sequences in 4k - 5k (bp) | 4800 - 5050 |
| 4th sequence | All merged sequences in 7k - 8k (bp) | 7500 - 7800 |
| 5th sequence | All merged sequences in 8k – 9k (bp) | 8450 - 8900 |

Given this, at this point, these are the CpG islands we believe to correspond to the real ones. Since our results don’t overlap perfectly, we can’t define clearly the bps at which the CpG island begins or at which it ends. Thus, those locations were determined by assuming reasonable values for its beginning and end, given the aforementioned data. Nevertheless, those two tools helped us to have an insight of generic region where those islands are.

The merged sequences from the CpG Islands that were not mentioned in the last table (1st sequence in 0-1k, all merged sequences in 3k-4k, all merged sequences in 5k-6k) for some reason don’t appear in the CpGPlot, even though we can’t understand clearly why, taking into account the algorithm this tool implements and statistics observed in CpG Islands (Obs/Exp ratio and %GC in each window).

**DNA Stats**

The DNA Stats tool returns the frequency of occurring nucleotides and of sequences of nucleotide pairs. Converted to probabilities (also available in this tool), the results are:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Followed by:** | | | |  |
|  | **A (%)** | **T (%)** | **G (%)** | **C (%)** | **Total (%)** |
| **A (%)** | 9.47 | 6.69 | 5.87 | 7.58 | 29.61 |
| **T (%)** | 7.47 | 7.58 | 4.25 | 6.17 | 25.48 |
| **G (%)** | 5.19 | 3.28 | 3.67 | 4.68 | 16.80 |
| **C (%)** | 7.48 | 7.92 | **3.02** | 9.98 | 28.10 |

The probability for the sequence pair CpG is the lowest from all possible sequence pairs (3.02%). If the given sequence was random then, considering that all nucleotides have the same probability of occurring and for large sequences as the one given to us to analyse (>10Kb), each sequence pair would have a probability of occurring around 25% x 25% = 6.25% (assuming that those two pairs are independent). The probability of CpG is less than half of this estimate due to the fact that cytosine is easily methylated and tends to mutate spontaneously into a thymine with a higher probability, through a deamination reaction. As a matter of fact, in the given sequence, we have P(G) = 16.80% and P(C) = 28.10% and, so, assuming again the independence of both nucleotides, we reach a probability around 4.72 %, which is still above the observed for the sequence pair CpG.

The analysis above was made considering the whole sequence and the always smaller than the expected observed frequency may suggest more than a simple randomness in DNA organization. In fact, those values can be explained by the aforementioned “decay” of methylated cytosines into thymines and the presence of CpG islands. Thus, one interesting point would be to apply this frequency-based probabilistic approach to a region that was previously identified as a CpG island to understand what happens on those regions. The results obtained for the region 2326-2546 were the following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Followed by:** | | | |  |
|  | **A (%)** | **T (%)** | **G (%)** | **C (%)** | **Total (%)** |
| **A (%)** | 11.82 | 6.36 | 4.55 | 7.27 | 30.32 |
| **T (%)** | 5.00 | 4.55 | 4.55 | 4.55 | 18.55 |
| **G (%)** | 7.73 | 2.73 | 8.18 | 4.55 | 23.08 |
| **C (%)** | 5.91 | 5.00 | **5.91** | 11.36 | 28.05 |

As it can be observed, while P(A) and P(C) remained almost constant, P(T) decreased around 7%, leading to an increase of the same magnitude for P(C). Moreover, we can observe that, for this CpG island, the probability for the sequence pair CpG increased almost 3%, being now significantly above the value calculated for the expected frequency of these sequences calculated for the whole sequence (4.72%), reinforcing the ideas previously discussed about these specific regions of DNA sequences.

**ORF Finder:**

This tool can be used to search for Open Reading Frames (ORFs), sequences that are divisible by three and bounded by stop codons, in a target DNA sequence, returning the range of each ORF, as well as its protein translation and, so, it allows us to search on a given sequence for potential protein encoding segments.

The parameters that can be selected in this tool are:

1. The codon which is at the beginning of the ORF;
2. The reading frame and the strand in which the search is supposed to be carried out;
3. The minimum number of codons (thus, amino acids) required for an ORF;
4. The genetic code to be used.

Regarding the last three points, they were considered fixed given the nature of our objective/work in this work: as far as reading frames are considered, we’re interested in any potential ORF and so the three reading frames were considered. An important point here is that both the direct and reverse strands were considered, since CpG sites are identified analyzing the sequence in the 5’ → 3’; the minimum number of codons considered were 30, since we wanted to find potential protein encoding segments and, for example, the smallest human protein has 44 amino acids, so at least has the same amount of codons. We considered 30 a reasonable value, giving us some margin. Nevertheless, we could observe that as we increase the minimum size required, the number of ORFs decreases since we’re being more restrictive in our selection criterion; finally, regarding the genetic code, we selected the standard one, since we don’t have any *a priori* knowledge about the organism we’re dealing with and, so, it is the only suitable option.

The results for the different accepted codons at the beginning of ORFs and strands are shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Direct Strand** | | | **Reverse Strand** | | |
| **RF 1** | **RF 2** | **RF 3** | **RF 1** | **RF 2** | **RF 3** |
| **ATG (ONLY)** | 11 | 6 | 11 | 10 | 5 | 15 |
| 28 ORFs | | | 30 ORFs | | |
| **ATG, GTG, CTG, TTG** | 15 | 14 | 20 | 17 | 15 | 21 |
| 49 ORFs | | | 53 ORFs | | |
| **Any Codon** | 40 | 30 | 37 | 34 | 25 | 31 |
| 107 ORFs | | | 90 ORFs | | |

From this table, as expected, as we become less restrictive regarding the starting codon, the number of ORFs increases. Moreover, from an analysis of the locations of the detected ORFs, the ORFs for a more restrictive beginning codon condition are all contained in some of the ORFs the less restrictive beginning codon condition. In fact, sometimes those ORFs do not coincide perfectly. In those times, the ORFs of the less restrictive beginning condition starts earlier due to the fact that there is an also accepted beginning codon before in the original sequence which is not accepted by the more restrictive condition.

Besides, there’s no relation between the number of bases and the strand on which we’re working (*i.e.*, for example, while for ATG (only) we have more ORFs on the reverse strand, for any codon we have more ORFs on the direct strand). Nevertheless, an interesting point is that the reading frame two is always the one with less ORFs for every situation., suggesting that there might be a cellular mechanism responsible for this reading frame being the less preferable. Between reading frames 1 and 3, we can’t say there’s a preferable one.

The sequence and the translated ORF were also outputs of this tool, but they were neglected since this information wasn’t our main focus.

Since usually the CpG islands are associated with the start of a gene (promoter regions – typically immediately adjacent to the gene in question, with a length of 100-1000 bp), we analyzed the location of the ORFs detected for the following conditions: ATG (only) as the beginning codon (more restrictive condition), using the three reading frames and the direct strand (since it’s the one of our sequence and where we’ve identified CpG islands with the previous tools), 30 as the minimum number of codons required and the standard genetic code (due to what was explained in the beginning of this tool exposition). The results are presented in the following table:

|  |  |  |
| --- | --- | --- |
| **Reading Frame 1** | **Reading Frame 2** | **Reading Frame 3** |
| 142 – 249  2329 – 2433  **2857 – 3132 (2)**  3415 – 3690  4021 – 4125  4138 – 4284  4612 – 4746  **5392 – 5544 (3)**  6955 – 7119  7189 – 7383  **8059 – 8268 (4)** | 1892 – 2050  **2651 – 2749 (2)**  6482 – 6574  6830 – 6919  6962 – 7054  8669 – 8797 | 1257 – 1352  **2439 – 2615 (2)**  3129 – 3302  3687 – 3851  4281 – 4703  4833 – 5102  6357 – 6452  7674 – 7802  **8964 – 9179 (5)**  **9231 – 9344 (5)**  9570 - 9725 |

The entries of the table that were pointed out (in bold) where the ones which allowed the existence of an CpG island before. In order to accomplish this relationship, ORFs whose beginning was not excessively after (300 bps at maximum) the end of the location of a given CpG island (see Table which merges information from CpG Islands and CpGPlot). The respective CpG island is referred by the number between parenthesis. In fact, only the first CpG island has no ORF detected after it. Thus, we discard the possibility of that region being a CpG island.

**b.**

The most homologous sequence found in the GenBank Nucleotide database was the mitochondrial DNA (complete genome) from *Abalistes stellaries* (Starry triggerfish) with a 100% match, (Score: 18484/18484, Query Cover: 100%).

Uma imagem com captura de ecrã, edifício

Descrição gerada automaticamente

In the figure above, it is represented the aforementioned DNA, as well as its different coding regions (horizontal bars) and the previously identified CpG islands for this DNA sequence (vertical bars, already neglecting the hypothesized CpG island in the region 470 – 735).

The first hypothesized CpG island (2300-2550) appears in the end of a gene coding ribosomal RNA, and so it might be placed in the promoter of the following gene (despite having a tRNA coding region immediately after, it can also be in the promoter respective to the gene ND1).

The second hypothesized CpG island (4800 - 5050) appears again in the end of a gene (ND2), but this time it has more than on tRNA coding regions after it. Nevertheless, it can also still be placed in the promoter of the gene COI, since sometimes promoters are not immediately before the gene that they “control”.

The third and four sequences hypothesized CpG islands (7500 – 7800 and 8450 – 8900, respectively) raise different problems: the third one occupies more than a half of the gene where it is placed (BAH10166.1), so it isn’t likely that it is in the promoter of the next gene; the fourth one, besides his excessive size in comparison to the gene to which it belongs (more than a 50% of it again), it is occupying the beginning of the next gene (COIII) as well. Given this, it is not likely that these two hypothesis are CpG islands in the promoter of genes.

Finally, still regarding the neglected hypothesized CpG island (470 - 735), even though it is in the end of a gene, it would occupy more than a half of the gene to which it would belong and, so, by the same token, it isn’t likely that it is a CpG islands in the promoter of a gene, as well.

# **Group iI**

**a.**

To prove the given expression’s validity, we must first show that

where is the length of the Markov Chain. Since is the transition probability from to which is given by then, the above sum is given by the sum of conditional probabilities over the same subspace . Therefore, the sum of all the possibilities conditioned to is .

Given this definition () then:

**b.**

The proof for this item depends merely on the observation that given a function

then

since is going to choose one over all the possible values to find the one that maximizes while the value of (and consequently ) is a constant which is not going to contribute for the maximization of .

In our case, the maximizing function can, indeed, be factorized by the conditional probability formula:

Hence, we have that:

**c.**

Here, we will be using the following probability axiom (:

And the following definition :

Therefore:

As a generalization, for any n-step transition from to , the probability of that transitions is:

# **Group iII**

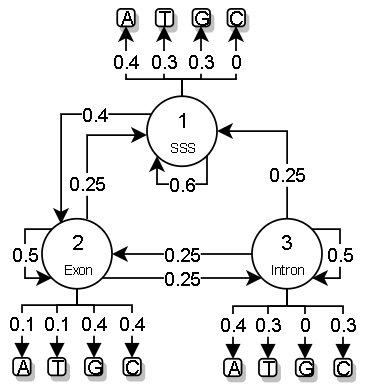
**a.**

The transition probabilities can be graphically represented using the following diagram, where the hidden states are represented by circles:

Uma imagem com relógio

Descrição gerada automaticamente

The emission probabilities can be added to the previous diagram where the emitted elements are represented by squares:



**b.**

The optimal path for sequence S is 211222221111112, which means that, from all the possible sequence of states that could have emitted sequence S, this is the one with the highest probability and therefore the most likely one. However, this sequence S may have been emitted by another sequence of states.

**c.**

P(S) = 9.38645997061689 e-10

To compute P(S), the Forward algorithm should be used, because it adds the probabilities for all possible paths that can generate sequence S.

This is done by applying iteratively (i) = P(,..., , = k) =

to each state for each letter of the sequence, starting in the first letter and in the end summing the probabilities for each state after emitting the last letter.

Adapting the algorithm means that instead of maximizing

**d.**

P() = 0.3593892076334204

P() = 0.64061079236658

P() = 0

P() = 0.4939335998198486

P() = 0.21385402172212853

P() = 0.2922123784580231

The posterior probabilities indicate that there is a higher likelihood of state 2 emitting the 4th letter because P() > P() > P() and that there is a higher likelihood of state 1 emitting the 9th letter because P() > P() > P().

In the optimal path, the state that emits the 4th letter is state 2 and the state that emits the 9th letter is state 1, which means that the most likely states to have emitted the 4th and 9th letters given sequence S correspond to the states that emit those letters in the optimal path. However, this is not always true because the optimal path depends on the maximization of a sequence of states and not just one state.