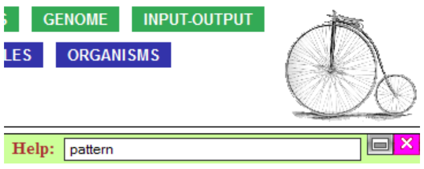
(17 April 2013)

MIC653: Advanced Molecular Genetics Bioinformatics & Computational Genomics

Patterns and motifs: Do it yourself

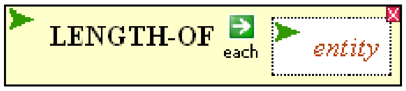
I. Familiarization with BioBIKE

Unless you are one of those rare individuals who were born knowing BioBIKE, you will undoubtedly need help in using it to your advantage. Here are ways of getting help:



- *Type a term in the Help box (as shown to the right and press Enter.*
- *Click a link on the resulting list of items to reach a help page*
- *Click a* **?** *associated with a function to reach a help page concerning that function*
- *Click* ***Help*** *on a function's Action Menu (its green wedge)*
- *Contact me at any time (ElhaiJ@VCU.Edu; 828-0794)*

I.A. Find the length of *Nostoc punctiforme* (the number of nucleotides in its genome)



1. Use the LENGTH-OF function in the Genome menu. Click the entity box. Find *Nostoc punctiforme* in the Data menu, N-Fixing-Cyanos submenu. Click Npun, the nickname of *Nostoc punctiforme*. Double click the name of the function (or click execute on the function's action menu).
2. What is the length of Npun's genome?

I.B. Find the average length of a gene of *Nostoc punctiforme*

1. Use the LENGTH-OF function in the Genome menu (or the Genes-Proteins menu).
2. Use the GENES-OF function in the Genome menu. Execute it to get a list of the genes of Npun.
3. Find the length of one gene. Copy and paste any gene in the list into the argument of LENGTH-OF. What is its length?
4. Find the lengths of all genes. You should get as many lengths as there are genes. If you get only one number, then the function returned to you the length of the list, not the length of each gene. To get the latter, specify EACH (by clicking the token arrow). This specifies that you want the length of each gene in the list and not the length of the list itself.
5. Run the lengths through the MEAN function (Arithmetic menu, Statistics sub-menu). What is the mean length of a gene in *Nostoc punctiforme*?

I.C. Display the upstream region of NpR3011 and NpF3012

1. Use the SEQUENCE-OF function in the Genome menu to display the chromosome of Npun. Then use the Go to box to go to the region of the chromosome containing NpR3011 (fill in the box with NpR3011 and click Go). What is the start codon of NpR3011? What is the gene upstream of NpR3011?
2. What are the coordinates of the region of the chromosome upstream of NpR3011?

3. Use SEQUENCE-OF and SEQUENCE-UPSTREAM-OF (Genes-Proteins menu Gene-neighborhood submenu) to display the sequence upstream of NpF3012. Compare it to the sequence you displayed in I.C.1. Find it?
4. Replace NpF3012 with NpR3011 to display the upstream sequence of the latter. Compare it to the display of the chromosome sequence. Find it?

II. Pattern matching to mine data

Rationale

Suppose you are interested in the interaction of bacteria and their phages. You realize that many phage are quite cozy with bacteria, integrating themselves into a bacterial genome for long periods of time. When a phage is integrated they're called prophages, and it can be difficult to distinguish their DNA from that of the native host, but this is the task you've set for yourself.

To learn how to do identify prophages, it would help to have on hand a set of known prophages on which you can try out your methods. Several prophages in E. coli are known, so your thought is to go there.

Throughout this exercise on pattern-matching you may find two sources of help useful, both accessible by putting the word pattern *into the* ***Help*** *box and pressing Enter. From the list that results, you'll be able to access a page called BioBIKE Pattern Matching and also the help page for MATCHES-OF-PATTERN.*

A. Search a Genbank file containing the annotation of *E. coli* for a specific text pattern

1. You could do this by going to NCBI, finding and downloading the file, and then uploading it into BioBIKE, but I've saved you the trouble. The annotation is available to you in a variable called `ecfile`.[*] You can also scroll through the file by mousing over the black **File** button and clicking **Shared files**. Then find the file called "Eschericiha coli-K-12-MG1655…" and click its name. Take a look at it. You'll need it for the rest of this question.
2. Search through the file until you find an annotation of a prophage. Don't pay much attention to gene-specific entries but rather find features that describe the coordinates of the prophage as a whole.
3. Figure out a pattern that will capture the beginning and end coordinates of each prophage along with the description of it (in quotes). Write out the pattern, including the quotation marks.
4. Use MATCHES-OF-PATTERN to extract all prophage information from the *E. coli* annotation. Be sure to use the CROSS-LINES option, since the information you seek may be on multiple lines. What output did you get?
5. Use DISPLAY-LIST to render in tabular form the information you extracted. What output did you get, and what might you learn from it?

[*] If you execute a function calling for `ecfile` and get the error message "`PROBLEM: I don't understand what you mean by 'ECFILE'`", then see the addendum at the end of this problem set.

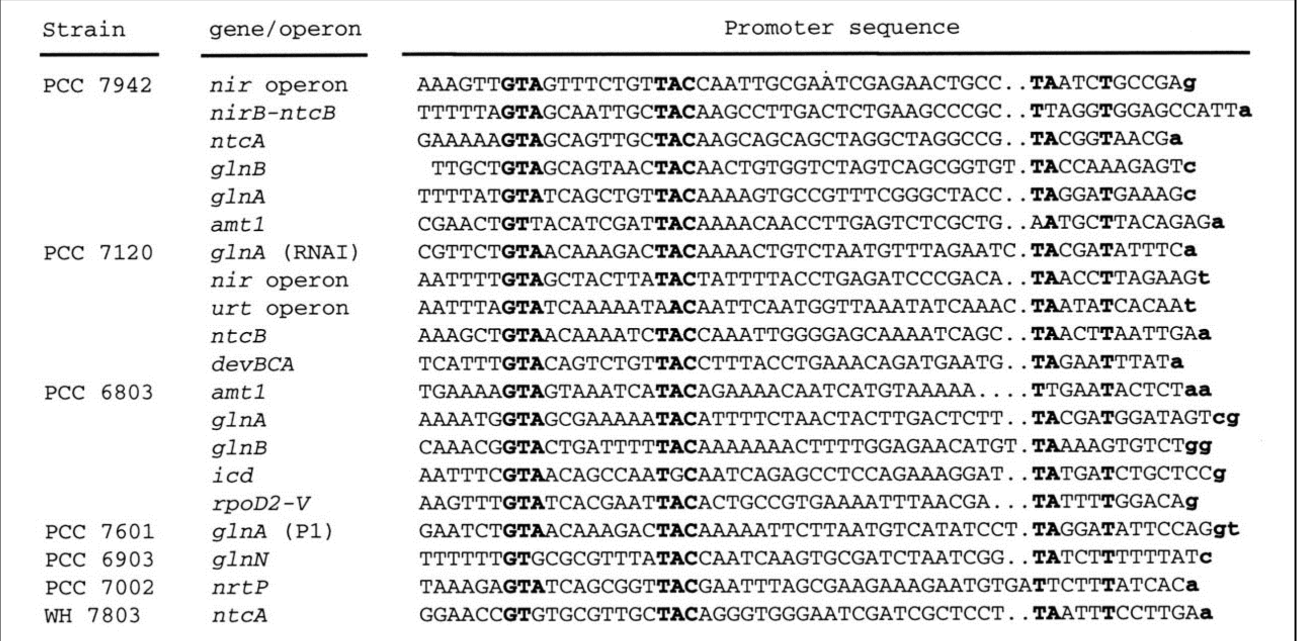
B. Search a protein sequence for a specific pattern of amino acids

Many redox proteins have iron-sulfur clusters as cofactors. In some cases, the binding of the cofactor to the protein depend on a precise constellation of cysteine residues, which are spaced Cys-X-X-Cys-X-X-Cys-X-X-X-Cys. Use this observation to identify proteins in *Synechocystis* PCC 6803 (nickname: S6803) that are candidates to be iron-sulfur proteins. To this end, the function PROTEINS-OF will be useful. In order to access the nominal annotation of the proteins you find, use DESCRIPTION-OF, with the EACH pre-option if you provide the function with a list of proteins, and the DISPLAY option for nice output. What proteins did you find? Which strike you from the annotation as likely iron-sulfur proteins?

III. Position-specific scoring matrices to find proteins with a given motif

Rationale

NtcA (for ***nit****rogen* ***c****ontrol) is a DNA-binding protein that responds to the availability of nitrogen, e.g. in the form of ammonia, to differentially regulate the expression of many cyanobacterial proteins involved in nitrogen metabolism. In several cases, the DNA sequence to which NtcA binds has been determined in the laboratory (see table below). You happen to be interested in a cyanobacterium, Anabaena variabilis (nicknamed Avar) for which there is no laboratory evidence concerning the binding of NtcA. This is an instance where bioinformatics may come to the rescue!*



Strain	gene/operon	Promoter sequence
PCC 7942	*nir* operon	AAAGTT**GTA**GTTTCTGT**TAC**CAATTGCGAATCGAGAACTGCC..**TA**ATC**T**GCCGA**g**
	nirB-ntcB	TTTTTA**GTA**GCAATTGC**TAC**AAGCCTTGACTCTGAAGCCCGC..**T**TAGG**T**GGAGCCATT**a**
	ntcA	GAAAAA**GTA**GCAGTTGC**TAC**AAGCAGCAGCTAGGCTAGGCCG..**TA**CGG**T**AACG**a**
	glnB	TTGCT**GTA**GCAGTAAC**TAC**AACTGTGGTCTAGTCAGCGGTGT.**TA**CCAAAGAGT**c**
	glnA	TTTTAT**GTA**TCAGCTGT**TAC**AAAAGTGCCGTTTCGGGCTACC..**TA**GGA**T**GAAAG**c**
	amt1	CGAACT**GT**TACATCGAT**TAC**AAAACAACCTTGAGTCTCGCTG..A**A**TGC**T**TACAGAG**a**
PCC 7120	*glnA* (RNAI)	CGTTCT**GTA**ACAAAGAC**TAC**AAAACTGTCTAATGTTTAGAATC.**TA**CGA**T**ATTTC**a**
	nir operon	AATTTT**GTA**GCTACTTA**TAC**TATTTTACCTGAGATCCCGACA..**TA**ACC**T**TAGAAG**t**
	urt operon	AATTTA**GTA**TCAAAAATA**AC**AATTCAATGGTTAAATATCAAAC.**TA**ATA**T**CACAA**t**
	ntcB	AAAGCT**GTA**ACAAAATC**TAC**CAAATTGGGGAGCAAAATCAGC..**TA**ACT**T**AATTGA**a**
	devBCA	TCATTT**GTA**CAGTCTGT**TAC**CTTTACCTGAAACAGATGAATG..**TA**GAA**T**TTAT**a**
PCC 6803	*amt1*	TGAAAA**GTA**GTAAATCA**TAC**AGAAAACAATCATGTAAAAA....**T**TGAA**T**ACTCT**aa**
	glnA	AAAATG**GTA**GCGAAAAA**TAC**ATTTTCTAACTACTTGACTCTT..**TA**CGA**T**GGATAGT**cg**
	glnB	CAAACG**GTA**CTGATTTT**TAC**AAAAAAACTTTTGGAGAACATGT.**TA**AAAGTGTCT**gg**
	icd	AATTTC**GTA**ACAGCCAA**TGC**AATCAGAGCCTCCAGAAAGGAT..**TA**TGA**T**CTGCTCC**g**
	rpoD2-V	AAGTTT**GTA**TCACGAAT**TAC**ACTGCCGTGAAAATTTAACGA...**TA**TTT**T**GGACA**g**
PCC 7601	*glnA* (P1)	GAATCT**GTA**ACAAAGAC**TAC**AAAAATTCTTAATGTCATATCCT.**TA**GGA**T**ATTCCAG**gt**
PCC 6903	*glnN*	TTTTTT**GT**GCGCGTTTA**TAC**CAATCAAGTGCGATCTAATCGG..**TA**TCT**T**TTTTAT**c**
PCC 7002	*nrtP*	TAAAGA**GTA**TCAGCGGT**TAC**GAATTTAGCGAAGAAAGAATGTGA**T**TCTT**T**ATCAC**a**
WH 7803	*ntcA*	GGAACC**GT**GTGCGTTGC**TAC**AGGGTGGAATCGATCGCTCCT..**TA**ATT**T**CCTTGA**a**

Binding sites of NtcA protein upstream from the promoter of several cyanobacterial genes. The sequences in bold are merely to draw to your attention relatively conserved nucleotides. The left hand portion is the NtcA-binding region and the right hand portion is the RNA polymerase binding region (i.e. the promoter). The table is from Herrero et al (2001) J Bacteriol 183:411-425.

A. Predict binding sites for a regulatory protein in an organism with no pertinent labarotory data

1. Use the sequences from this table to find possible NtcA-binding sites in Anabaena variabilis ATCC 29413 (nickname Avar). Investigate at least the first few matches to determine if the annotation is suggestive of a role in nitrogen metabolism. You'll be interested in the following functions:

APPLY-PSSM-TO
DESCRIPTION-OF

You will be relieved to know that the sequences are available to you (no need to type) by using the variable `ntcA-sites`.[*] Note that the *with-pssm-from* box of APPLY-PSSM-TO requires a list of sequences (e.g. `ntcA-sites`). You might reasonably think that it takes an actual PSSM, such as that produced by the MAKE-PSSM-FROM function, but it doesn't. What are the top matches and what did you conclude? What code did you use?

2. Determine the information content of the aligned sequences. Based on what you find, consider altering your approach to III.A1. These functions will be helpful:

   INFORMATION-OF
   PLOT

   Note that, like APPLY-PSSM-TO, INFORMATION-OF requires a list of sequences (e.g. `ntcA-sites`). Again, a PSSM won't work. What does a plot of the information content look like? What code did you use to get the plot? How did you make use of your findings?

3. The table shown above has variable gaps in the alignment to make both parts of the sequence align. PSSM's don't do well with sequences with variable gaps. Consider ways in which you could make use of the full information in the table. Ideas?

B. Troubleshooting through patterns.

When I first put in the sequences shown in the table, I made some typographical errors. You can verify this by using my original effort (available to you as `ntcA-sites-old`)[*] in III.A. You can stare at the sequences (as I did) by displaying them using DISPLAY-LIST with the EACH pre-option, but you'll probably have better luck if you use pattern matching to detect the problem. What pattern can you use to find the characters in the sequences that are ***not*** legitimate nucleotides? The pattern cheat sheet on the course web site might help. Ideas?

IV. What DNA pattern marks the beginning of protein-encoding genes?

Rationale

*Since the title of today's module is "*Pattern Recognition and Gene Finding*", I feel compelled to end with something related to finding the pattern beginning a gene. How does the cell plow through huge stretches of nucleotide sequences to recognize where a gene should begin? You no doubt have your own ideas, but put them aside for the moment and try the following.*

Define a set of sequences upstream from the protein coding genes of your favorite organism or bacteriophage. Consider only the 20 nucleotides upstream. The following functions will be of use:

[*] If you execute a function calling for `ntca-sites` and get the error message "`PROBLEM: I don't understand what you mean by ...`", then see the addendum at the end of this problem set.

SEQUENCE-UPSTREAM-OF (using the LENGTH option)
CODING-GENES-OF
MOTIFS-IN (using the DNA option)

If you choose a bacterium, you'll probably run out of time, hitting the execution time limit of 40 seconds. Not a problem! You don't need thousands of upstream sequences to find a good motif. One solution is to take a random sample of the upstream sequences, using the following function:

CHOOSE-FROM (using the WITHOUT-REPLACEMENT and TIMES options)

200 samples should be plenty and will drastically cut down the execution time. What motifs did you find? Using what code? What is the significance of the motif(s)?

V. Forensic applications

Short tandem repeats (STRs) are regions of DNA consisting of a unit 3 to 6 nucleotides in length repeated multiple times, for example AGGTAGGTAGGTAGGT…. Since the number of repeating unit mutates frequently, relative to other DNA changes, they are commonly used as a source of variation to identify individuals. They are also important as the causative agent of some genetically determined diseases.

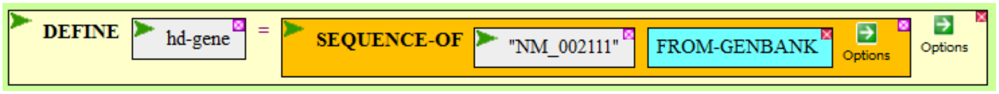
A. Recognizing a specific STR

1. MATCHES-OF-PATTERN can be used to recognize STRs. To try it out, use this function to find the full extent of the repeated AGGT unit in:

   TCCTTCAGGTAGGTAGGTAGGTCCGCCA

   Bring down MATCHES-OF-PATTERN from the **All** alphabetical menu. Copy/paste the above sequence into the *target* box, putting it between " " to identify it as a literal string rather than a variable name. In the *pattern* box, enter a string (between " "). That string should consist of the repeated unit as a group contained within ( ) repeated an indefinite number of times. Thus, the group would be (AGGT) . See the list of BioBIKE pattern matching symbols to find the symbol for indefinite repetition.

2. Huntington's disease is caused by an excess of repeats of CAG (encoding glutamine) in the *HUNTINGTIN* gene. Individuals with fewer than 28 repeated units have a normal phenotype, while those with greater than 36 express some symptoms of Huntington's disease. Obtain the sequence of the gene from some individual using the SEQUENCE-OF function with the FROM-GENBANK option, as shown below:



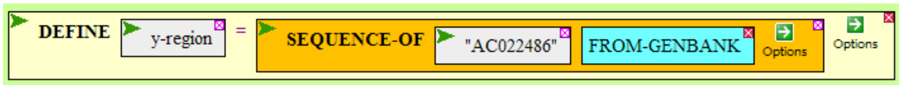
```
DEFINE hd-gene = SEQUENCE-OF "NM_002111" FROM-GENBANK
```

   Use MATCHES-OF-PATTERN to identify the full extent of the CAG repeat in `hd-gene`. Finally, use the coordinates reported by this function to find the CAG repeat in the sequence of the gene, displayed with SEQUENCE-OF `hd-gene`. What do you predict is the phenotype of the person carrying this gene, what evidence do you have for that prediction, and what code did you use to find it?

B. Identifying unknown STRs

You have sequenced a portion of the Y-chromosome from an individual and want to determine if there are any STRs in the segment.

1. Obtain the sequence from GenBank by executing the function shown below:



DEFINE y-region = SEQUENCE-OF "AC022486" FROM-GENBANK Options Options

2. Use MATCHES-OF-PATTERN to find all instances of 3 to 6 undetermined characters repeated at least 6 times. To specify a repeat of a previously captured group, you need to resort to a new symbol: `n, where n is a number. `1 refers to the first group captured, `2 refers to the second, and so forth. So the pattern "(Wa*...)-`1" would match "Walla-Walla", since "(Wa*...)" would capture "Walla" and be named group 1. What is the longest STR and what code did you use to find it?

VI. Gene/genome comparisons through oligonucleotide contrasts

Genomes are more or less homogeneous with respect to nucleotide and oligonucleotide composition, at least averaged over large chunks. This characteristic can be used to identify DNA fragments of sufficient length as typical of one organism or another. Here's a case in point.

Suppose that you've isolated from the Chesapeake Bay a small, DNA-containing particle, which you suspect is a virus. But it came with no informational tag... what does it infect? Since by electron microscopy, it is seen to be proteinaceous, it's probably a bacteriophage, and since the most prevalent bacteria in the ocean are cyanobacteria, your guess is that it infects one of them. Here's a rogue's gallery of marine selected cyanobacteria:

Crocosphaera watsonii (nicknamed cwat)
Prochlorococcus marinus ss120 (nicknamed ss120)
Synechococcus WH8102 (nicknamed S8102)
Trichodesmium erythreum (nicknamed ter)

You would think that the genome characteristics of a phage would be similar to those of its host. Presuming this is the case, then your strategy is to compare the dinucleotide frequencies in the phage genome with the dinucleotide frequencies in the four candidate host cyanobacteria.

A. Make a tool to calculate dinucleotide frequencies

1. First teach yourself how to do it, then teach the computer. Count just one dinucleotide (say "AA") in just one genome (say W8102). To do this bring down the COUNT-OF function from the **Strings-Sequences** menu, **String-Analysis** submenu. In the *query* box, type "AA" (with the quotation marks) and press Enter. Choose the **IN** option, and click **Apply**. Enter S8102 for the *value* of the IN option (not in quotes, because it represents the organism – it isn't the literal letters W-8-1-0-2). Execute the function. You should see a very large number in the Result pane at the bottom.

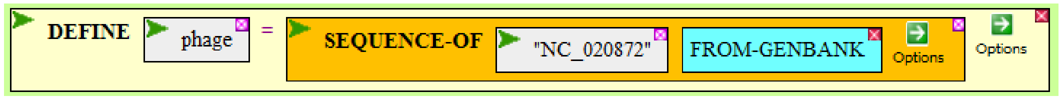
2. Now to generate all 16 dinucleotides. Bring down ALL-DNA-SEQUENCES from the **Strings-Sequences** menu, **String-Production** submenu. Fill in the *given-*

length box with 2. Execute the function to convince yourself that it is working as expected, producing the 16 dinucleotides.

3. Now learn how to count the 16 of the dinucleotides. Delete "AA" (using the small x in the upper right corner of its box), and replace it with the ALL-DNA-SEQUENCES function from **VI.A.2**, either dragging that function into the box or cutting/pasting. Execute COUNT-OF again... Error. Whoops! COUNT-OF is not very intelligent, you need to tell it that you want a COUNT-OF EACH element of the list. Select the EACH option next to COUNT-OF and try executing again. In a few seconds you should get a list of 16 large numbers.
4. To calculate the frequency, you need to divide all of these numbers by the total number of nucleotides, i.e. the length of S8102. Bring down the QUOTIENT-OF function (from the **Arithmetic** menu). Drag or copy/paste the COUNT-OF function into the numerator *number* box. Then fill the denominator *number* box with LENGTH-OF (from the **Genome** menu, **Sequence-Analysis** submenu). Execute the function to get a list of 16 fractions.
5. Now that you know how to calculate dinucleotide frequencies, teach BioBIKE how. To do this, bring down DEFINE-FUNCTION from the **Definition** menu. Provide a *name* for the function (perhaps **dinucleotide-frequencies** or some such – just so long as there are no spaces in the name). Drag or copy/paste the QUOTIENT-OF function into the *form* box of the body of the DEFINE-FUNCTION form. Now to generalize the calculation. You were using S8102 for testing purposes. Now replace both instances of s8102 with a variable (perhaps `organism`). Whatever you call that variable, put the same name in the *arg* box of the function (*arg* is short for *argument*). Finally, execute the DEFINE-FUNCTION. If all is well, a new button should appear at the top of the screen, giving you a **Function** menu, and your new function should be on it.
6. Test the new function by bringing DINCULEOTIDE-FREQUENCIES down from the **Function** menu and providing S8102 as the argument. Execute the function. You should get the same list of frequencies as before.

B. Compare the phage dinucleotide frequencies with those of the candidate hosts

1. You can now readily calculate the dinucleotide frequencies of the four candidate hosts by using your new tool, simply by replacing the argument with the different nicknames of the cyanobacteria. Try it.
2. The phage is more of a problem, since CyanoBIKE doesn't intrinsically know the phage sequence. Get it from GenBank as follows:



```
DEFINE phage = SEQUENCE-OF "NC_020872" FROM-GENBANK Options Options
```

Build and execute that function. Then use `phage` as the argument to your DINUCLEOTIDE-FREQUENCIES function.

3. You can now determine which of the four sets of cyanobacterial frequencies is closest to the set of phage frequencies. One way is to subtract each element of the phage set from the corresponding element of the bacterial set (using DIFFERENCE-OF), taking the absolute values of the list of differences (using

ABS), and summing those absolute differences (using SUM-OF). Try it. Which bacterium gives the least total difference?

4. ...or you could try the built in method, using the DINUCLEOTIDE-COMPARISON function (**Genome** menu, **Sequence-Analysis** submenu), using the phage as *entity1* and each of the bacteria as *entity2*. Which bacteria gives the least total difference?

VII. Identifying anomalous oligonucleotide frequencies in genomes

Genomes are not random (if they were, we wouldn't be having this conversation). It's long been recognized that when confronted with a code you don't understand (e.g. the human genome, to a large extent), a useful strategy is to catalog those elements that appear surprisingly frequently or surprisingly infrequently.[*] *In this section, you'll look for rare 6-nucleotide sequences in a much smaller genome, that of the cyanobacterium Anabaena PCC 7120 (also called Anabeana variabilis ATCC 27893). That way, you'll be able to finish the job in a reasonable length of time. Furthermore, also to save time, I will suggest that you confine yourselves to just those 6-nucleotide sequences consisting of C's and G's. That cuts down the number from 4096 (4^6) to 64 (2^6) and will greatly speed up the execution of your program.*

A. Calculate the frequencies of all CG-containing hexanucleotides in *Anabaena* PCC 7120

1. Use COUNT-OF to obtain a count of a specific hexanucleotide sequence (say, CCCCCC) in A7120. Bring down the COUNT-OF function from the **Strings-Sequences** menu, **String-Analysis** submenu. In the *query* box, type "CCCCCC" (with the quotation marks) and press Enter. Choose the **IN** option, and click **Apply**. Enter A7120 for the *value* of the IN option (not in quotes, because it represents the organism – it isn't the literal letters A-7-1-2-0). Execute the function.
2. That's one of the 64 hexanucleotides,... how to get the rest? Bring down the ALL-COMBINATIONS-OF function (**String-Sequence** menu, **String-Production** submenu). Fill in the *alphabet* and *given-length* boxes and execute the function. Does it give you the 64 hexanucleotides consisting solely of C's and G's?
3. Now that you know how to count a single hexanucleotide, you'll apply this function to all 64 CG-containing hexanucleotides. To do this, bring down the APPLY-FUNCTION (**Definition** menu). Drag or copy/paste the function from **VII.A.1** into the *function* box. Replace "CCCCCC" in that function with some variable, named what you will (e.g. oligo or seq or...). Whatever you called it, but that variable in the *variable* box to tell the function what will vary. Finally, drag or copy/paste the ALL-COMBINATIONS-OF function from **VII.A.2** into the *list* box. Execute (this will take several seconds)...
4. Whoops! You get all these numbers, but which oligonucleotides are they counts of? We need to associate the name of the oligonucleotide with the count. To do this, go back to the COUNT-OF function and, from its action menu (the green wedge to the left of COUNT-OF) select **Surround With**. That will select the function and allow you to wrap a LIST function around it, chosen from the **List** menu. The **More** menu of LIST will enable you to add another item. Fill the resulting box with oligo (or

[*] Example: *The Gold Bug* by Edgar Allen Poe.

whatever you named the variable. Now instead of generating numbers, you'll generate lists, each list consisting of a number and the oligonucleotide. Execute the function.

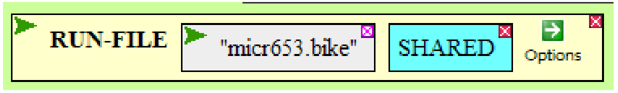
5. Better, but it would be nice to have a sorted list. Wrap the entire APPLY-FUNCTION (by selecting **Surround With** again) with the SORT function, found from the **List-Tables** menu, **List-Production** submenu. Execute this.
6. Finally, to get a nicely formatted list, bring down DISPLAY-LIST (from the **Input-Output**) menu and drag or copy/paste the result into the *list* box, and execute the function. **Provide a copy of the final result and the code that produced it.**

B. Biological significance of the hexanucleotide frequencies

1. I imagine that you are marveling at the result from **VII.A.6**. What an enormous range of values! What property can you detect in the six most frequent hexanucleotides? What biological explanation can you put forward for their high frequency?
2. And what about the five least frequent hexanucleotides? What property do they share? This property might not be evident to you unless you write out the hexanucleotide as double-stranded DNA.
3. What could be the cause of this extreme deficiency in hexanucleotide sites? One clue can be found by going to REBASE (http://rebase.neb.com/), a database of restriction enzymes and DNA methyltransferases. In the Search Category box, choose Organism Name, enter Anabaena, and click Go. You should find in the resulting list a link for Anabaena variabilis ATCC 27893. Following it gets you to a list of restriction enzymes and methyltransferases found in the organism. There are only four restriction enzymes (not preceded by M(ethyltransferase) or followed by P(artial)). Only one recognizes a sequence consisting solely of G's and C's (recognizing that Y=T or C; R=A or G; W=A or T). Which is it? How does it relate to the list of hexanucleotide frequencies? Any theories as to why the infrequent hexanucleotides are infrequent?

Addendum: What if pre-made variables don't work?

If you attempt to use ecfile, ntca-sites, or ntca-sites-old (properly spelled!) and get the error message "`PROBLEM: I don't understand what you mean by…`", *it could be the variables have disappeared. To get them back, go to the* ***All*** *menu, bring down RUN-FILE, and execute the function as shown to the right.*


RUN-FILE
"micr653.bike"
SHARED
Options