MIC653: Advanced Molecular Genetics Bioinformatics & Computational Genomics
Projects: Pattern-based search for half-remembered gene involved in aging

Search for half-remembered gene

(use StreptoBIKE)

Rationale

I don't know how many times a week I remember an interesting article – sort of – and think, "Where did I see that?" If I happen to remember to remember an author's name (an author not named Zhang) or some other relatively unique identifier, I have a chance of rediscovering the article by a search of PubMed. But I'm often not that lucky.

Consider a recent case, I recalled – sort of – an article about a gene whose product was critical in the aging of yeast. The absence of its gene product halved the lifespan of yeast, and its presence was required for caloric restriction to increase lifespan. A pretty important gene! …What was it again?

All I could dredge up from my memory was that it was a relatively recent article (later than 2016) and the gene started with an "S". No authors, no key words (apart from "yeast" and "aging"). So all I could do was take what I had to PubMed and hope for the best.

Here's what PubMed was able to offer:

History — Download history | Clear history

Search	Add to builder	Query	Items found	Time
#3	Add	Search **(("2017/01/01"[Date - Publication] : "3000"[Date - Publication])) AND #2**	683	10:22:36
#2	Add	Search **(#1 AND aging)**	3480	10:21:51
#1	Add	Search **(yeast OR saccharomyces)**	287635	10:21:37

Statistics from a search of PubMed for articles related to aging of yeast, yielding 683 candidates.

683 articles exceeded my tolerance for reading abstracts, let alone full text. What to do? I didn't use perhaps my biggest clue – the gene starts with an "S", but how could I do that?

Solution: A computer plus a human to program it.

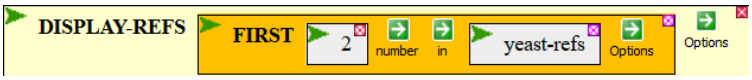
Throughout this exercise on pattern-matching you may find two sources of help useful, both accessible by putting the word `pattern` into the **Help** box and pressing Enter. From the list that results, you'll be able to access a page called `BioBIKE Pattern Matching` and also the help page for MATCHES-OF-PATTERN. The in-class presentation may also be of some use to you. If you haven't used BioBIKE before, it might be a good idea to invest some time learning about how it works. You can do this through a short on-line tutorial available here or through the portal.*

* Once at the portal (http://biobike.csbc.vcu.edu), click the link to the *guided tours*, and choose the tour called *BioBIKE syntax and conventions.*

A. Load into StreptoBIKE the references a nd a useful function

I downloaded the 683 candidate articles in a form amenable for computation and put the file in the shared directory of StreptoBIKE. I also created a function to display them in a readable format (plus another one described later). Here's how you can see the display function in action:

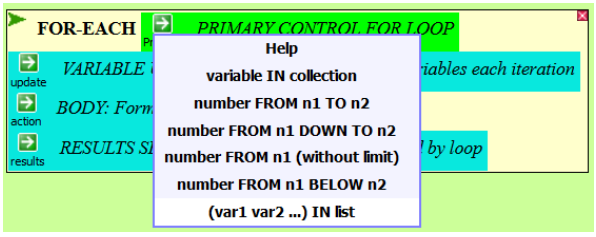
- Go to StreptoBIKE.
- Under the **File** button, click **User contributed stuff**
- Click Use this package next to **Yeast-refs**. That will bring into your **Variables** button the variable `yeast-refs` (containing the 683 references) and bring into your **Functions** button a function `DISPLAY-REFS` that will show you references from the list in a readable format.
- Mouse over the **Functions** button and click `DISPLAY-REFS`.
- Click the *refs* box and bring into it the function `FIRST` from the alphabetical list found under the **All** button.
- Click the *number* icon of `FIRST` and enter the number 2.
- Click the *entity* box and bring into it the variable `yeast-refs` from the **Variables** button. You should end up with the following function:



- Execute this function (**Execute** can be found by mousing over the green wedge to the left of `DISPLAY-REFS`). You should see displayed in a popup window the first two references.

B. Search PubMed output for those with genes starting with "S"

To look for references with genes starting with "S", you'll write a program that considers each of the 683 articles in turn, searching each abstract for a word beginning with "S" and fulfilling the standard format of yeast genes ($$$#..., where $ represents a capital letter and #... *represents* one or more digits). A `FOR-EACH` loop will accomplish the repetitive slog through the articles, and `MATCHES-OF-PATTERN` will look for the gene.

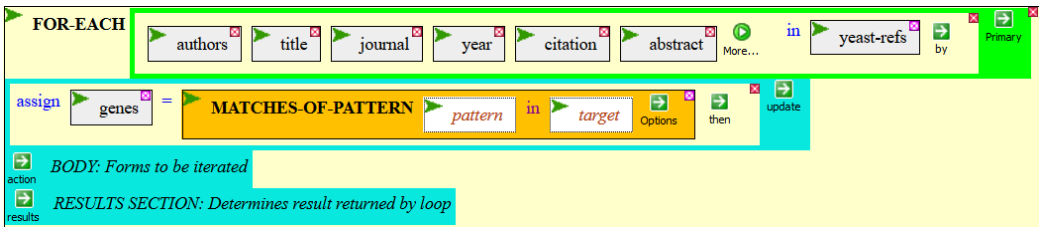


Bring down a `FOR-EACH` loop from the alphabetical list and click the icons to the left of *Additional controls*, *Initialization*, and *Final Action*. In each case click **Hide**, because you won't be needing these functionalities. Click the icon to the left of *Primary control*, and click **(var1 var2…)**, as shown to the right. This option enables you to give names to each part of a reference, as described below.

Mouse over the More icon to the right of the *var* box and select **Add another** and **Add two more** sufficient times to get a total of six *var* boxes. Fill each *var* box with one of the following, in order: `authors`, `title`, `journal`, `year`, `citation`, and `abstract`. Fill the *list* box with `yeast-refs`, found by mousing over the **Variables** button.

This will give you the raw material needed to do the search. To do the search and save the results, mouse over the icon to the left of *Variable update* and click **ASSIGN variable = value**. The *initial value* box can be satisfied by getting MATCHES-OF-PATTERN from the

alphabetical list of functions. I suggest you save the result of the search performed by this function in a *var*iable called `genes`. Here's what you should have so far:



MATCHES-OF-PATTERN will produce a list of genes if any are found or nothing if not. The *Results-section* of the loop should save the list if it exists. Mouse over the icon to the left of this section and click **when… append**. Set the *condition* to `genes` and the *value* also to `genes`. This signifies that when `genes` exist (when a list of at least one gene was found) then that list will be appended to the collection of found genes.

All that's left is to fill in *target* and *pattern* boxes of MATCHES-OF-PATTERN. The simplest approach is to search just the abstract, putting `abstract` into the *target* box. I leave the pattern to you. You'll probably need to experiment to get it right. The pattern should begin with the letter "S" and continue with the rest of the general format for yeast genes. Finally, mouse over the **Options** icon and click **+Matches** to tell the function you just want the genes, not their coordinates within the abstract.

Execute the function to see what you get (don't worry about the warnings generated about unused variables).

Possible problems. You might get words that don't fit the format for a gene name, such as "Q157" (has 1, not 3 letters). If so, look carefully at how you specified your pattern and exactly what the specifications mean, according to the description of BioBIKE Pattern Matching symbols (see Rationale).

A second problem is that a reference may name a gene several times. You can get rid of duplications by mousing over the icon to the left of the *Body* section, clicking **body**, and filling the *form* box with this:

Now try executing the function again. If all is well, you will have a list of genes found in the 683 abstracts. What are the most popular genes in these articles (maybe that will jog your memory). Use the function COUNT-ELEMENTS-OF to find out. Bring down the function from the **Functions** menu, copy and paste the result you just got into the *list* box, and execute the fnnction.

What are the most popular genes in yeast aging articles?

Modify your search so that when a genes list exists you collect the name of the journal instead of appending the genes list. You will need to replace **when…append** with **when…collect**, because you can append only lists (e.g. lists of genes), not single items (e.g. a journal name).

What are the most popular journals in yeast aging research?

C. Search PubMed output for those articles studying genes starting with “S”

Now that you remember what was the mystery yeast gene (and if you don’t, presume it was the most popular gene), modify the pattern used by MATCHES-OF-PATTERN to specify that exact gene, and modify the *Results* section by collecting the references in which the gene is found. Do this with **when…collect** that looks something like this:

(you can put whatever elements of the reference you want displayed, e.g. the abstract).

Then copy/paste the result into the DISPLAY-REFERENCES function and reap your reward.