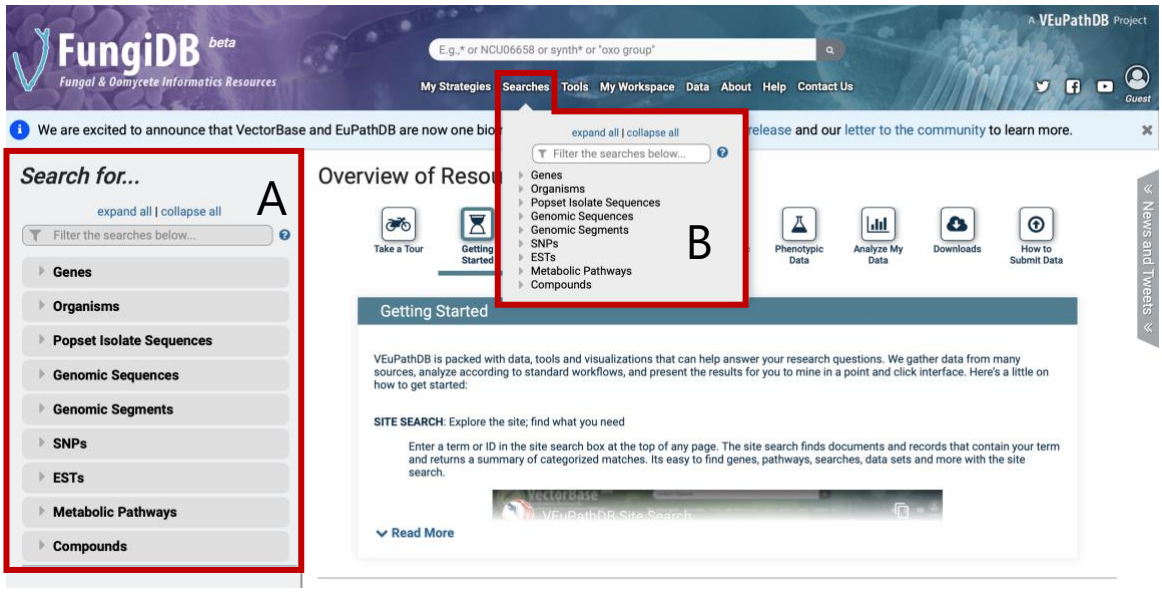
Regular Expressions & Genomic Colocation

Note: *this exercise uses different VEuPathDB resources as example database, but the same functionality is available on all VEuPathDB resources.*

Learning objectives:

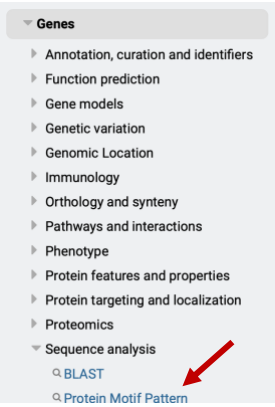
- Run a regular expression search on amino acid sequences
- Run a regular expression search on nucleotide sequences
- Use the genomic colocation search

Protein or nucleotide sequences can be identified using the regular expression searches in VEuPathDB. This search is very useful to identify patterns of sequences.
Searches can be accessed from categorized menus in the left search for panel (A) or from the searches menu in the header (B).



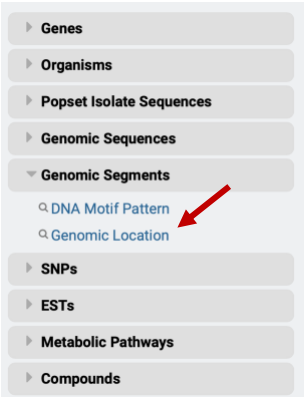
Accessing the protein motif pattern search:

- Click on the *Genes* category then click on the sequence analysis category



Accessing the DNA motif pattern search:

- Click on the *Genomic segments* category



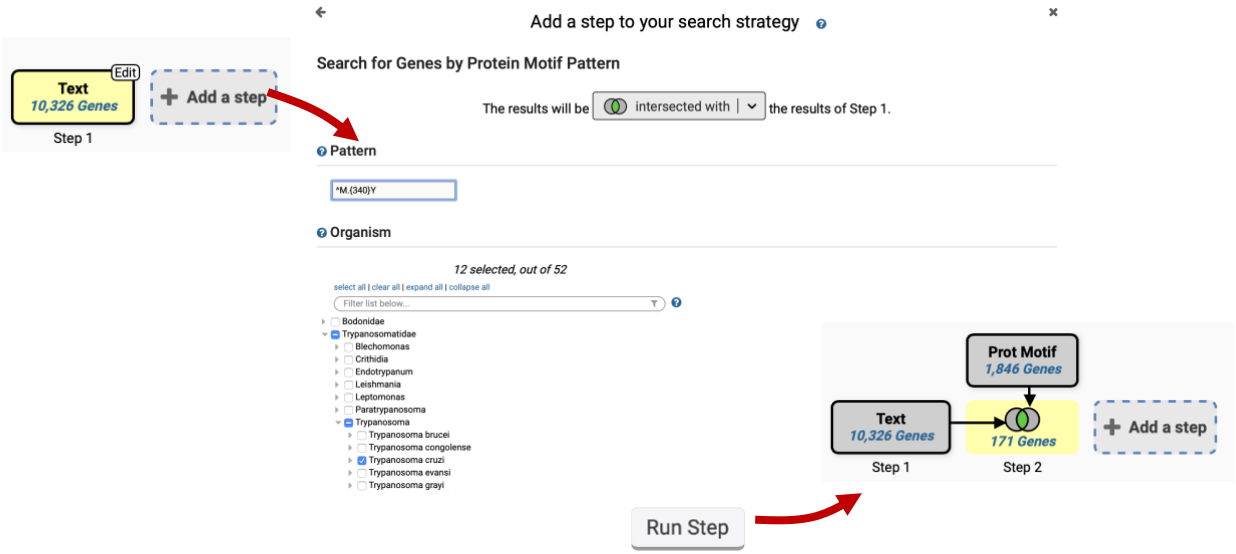
Note: **the appendix at the end of this document includes additional regular expression help.**

1. Using regular expressions to find motifs in TriTrypDB: finding active trans-sialidases in *T. cruzi* (TriTrypDB).

a. *T. cruzi* has an expanded family of trans-sialidases. In fact, if you run a text search for any gene with the word “trans-sialidase” in its *product description*, you return over 10000 genes among the strains in the database!!! Try this and see what you get.

b. Not all of the genes returned in (a) are predicted to be active. It is known that active trans-sialidases have a signature tyrosine (Y) at position 342 in their amino acid sequence. Add a motif search step to the text search in ‘a’ to identify only the active trans-sialidases.

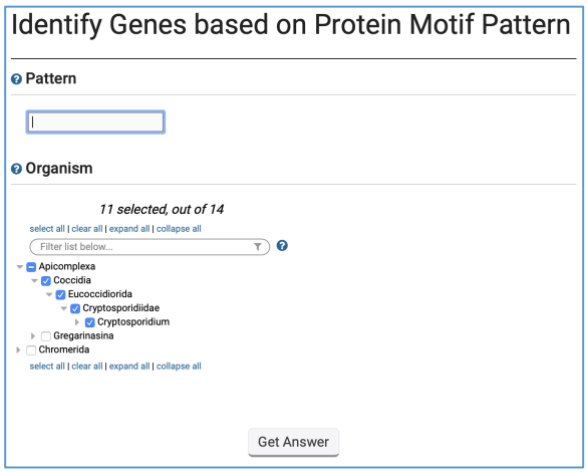
c. Write a regular expression that defines a protein sequence that starts with a methionine, and is followed by 340 of any amino acids, followed by a tyrosine ‘Y’.



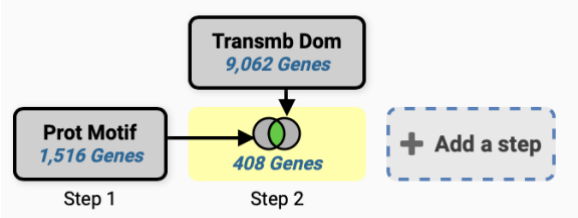
2. Find Cryptosporidium genes with the YXXΦ receptor signal motif (CryptoDB)

The YXXΦ (Y=tyrosine, X=any amino acid, Φ=bulky hydrophobic [phenylalanine, tyrosine, threonine]) motif is conserved in many eukaryotic membrane proteins that are recognized by adaptor proteins for sorting in the endosomal/lysosomal pathway. This motif is typically located in the c-terminal end of the protein. ****Note**: *do not look for the Φ symbol on your keyboard – this will not work. Rather you should use the amino acid symbols.*

a. Use the “protein motif pattern” search to find all *Cryptosporidium* proteins that contain this motif anywhere in the terminal 10 amino acids of proteins. (hint: for your regular expression, remember that you want the first amino acid to be a tyrosine, followed any two amino acids, followed by any bulky hydrophobic amino acid (phenylalanine, tyrosine, threonine).



b. How many of these proteins also contain at least one transmembrane domain.



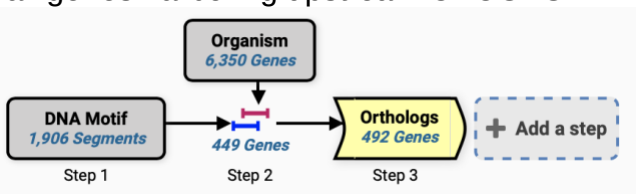
c. What would happen if you revise the first step (the motif pattern step) to include genes with the sorting motif in the C-terminal 20 amino acids? (hint: edit the first step and modify your regular expression).

Note: if you need help with the regular expression the answers are in appendix B.

3. Find fungal genes downstream of a regulatory DNA motif (FungiDB).

Transcriptional start sites are often located within a certain distance upstream of the genes or gene clusters that they regulate. In fungi, DNA motifs are also important for regulation of processes linked to host cell invasion or production of secondary metabolites. Readily available genomic data facilitate the discovery of regulatory motifs via examination of orthologous sequences.

The goal of this exercise is to identify all genes harboring upstream CACGTG motif, known for its role in transcriptional regulation. We will start our search in an extensively studied model organism *Saccharomyces cerevisiae* and expand our search to *Fusarium graminearum*.

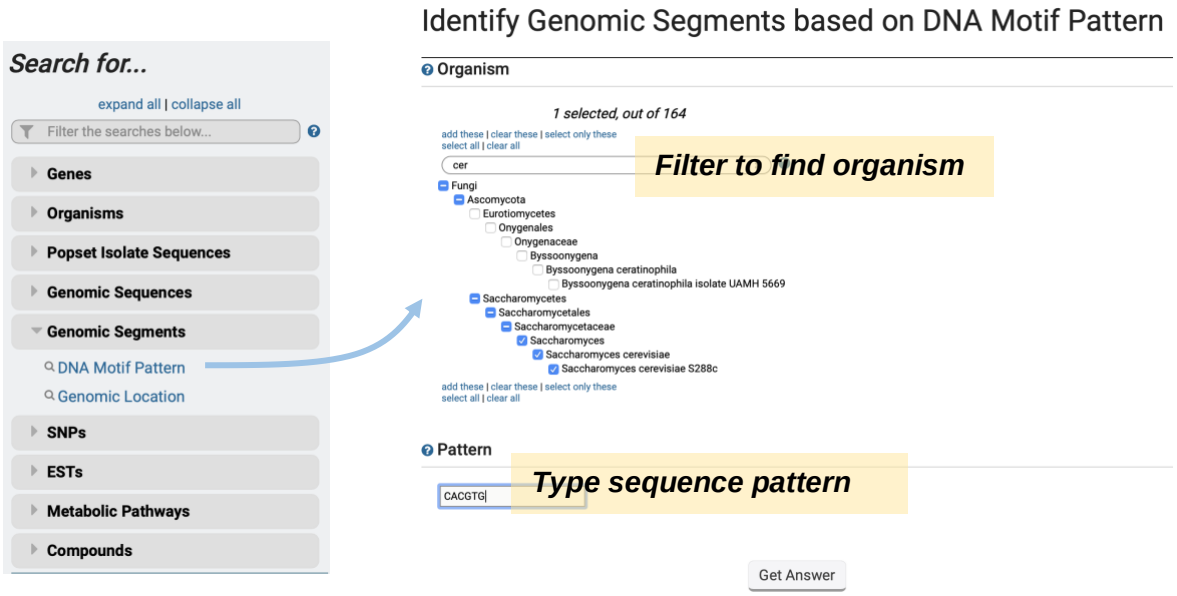


Here is a summary of the search strategy:

a. Find the CACGTG DNA motif in the *Saccharomyces cerevisiae* genome.

1. Select the “Search for genomic segments (DNA motif)” menu from the Search menu and look for CACGTG in *S. cerevisiae*.

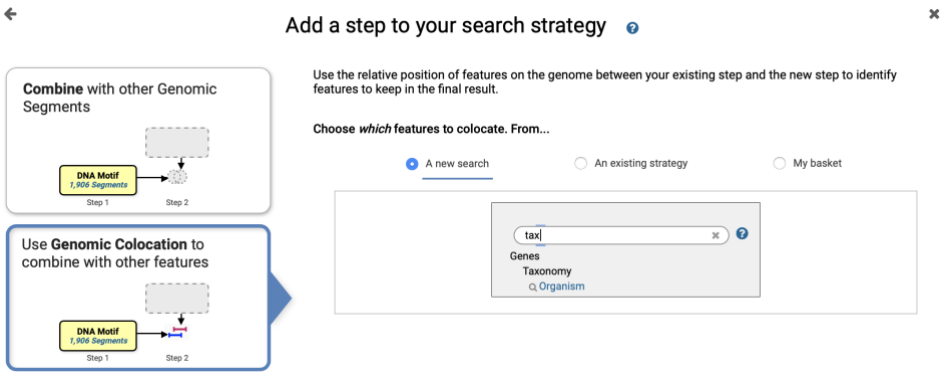
2. Your search returns over 1900 DNA segments containing GACGTG motif. Next, let's look for putative regulatory targets of this motif by searching for genes that are located 600bp downstream of this sequence.



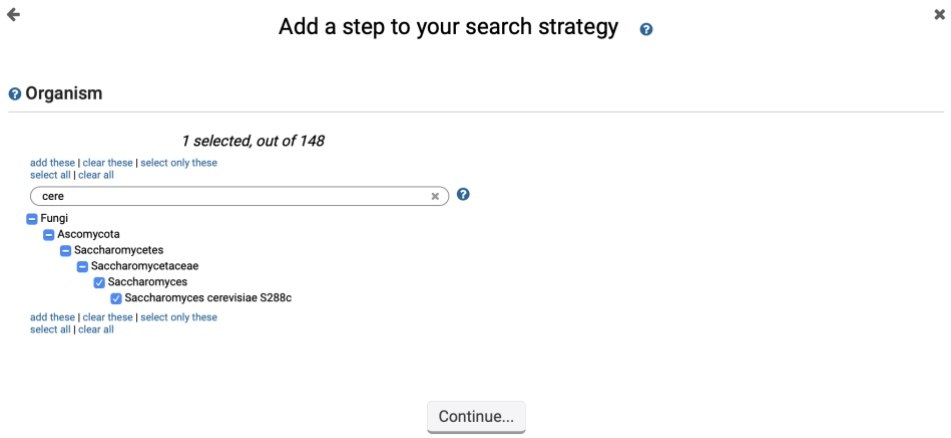
b. Identify genes with the CACGTG motif located 600bp upstream of an open reading frame.

EuPathDB offers a colocation function to identify genomic features within a specified distance of each other. Run a search for all genes in *Saccharomyces cerevisiae* and use the colocation tool to identify genes that contain the CACGTG motif in their upstream regions. Follow these steps:

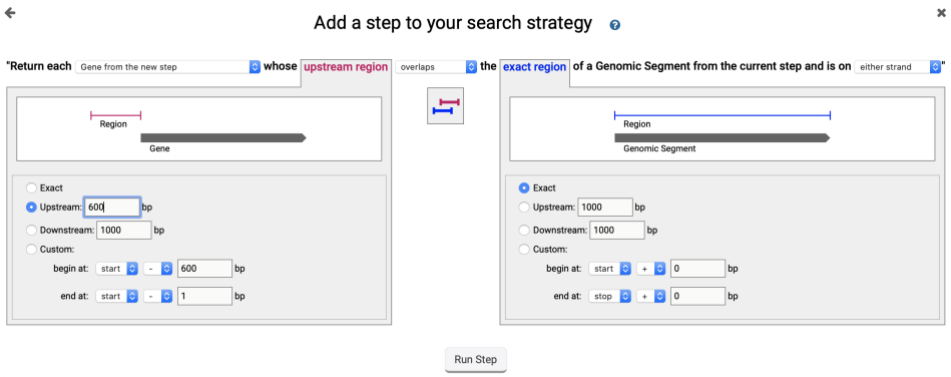
1. Click "Add Step". Choose the option on the left called "Use **Genomic Colocation** to combine with other features" then select the *organism* gene search which can be found under the *Taxonomy* category.



2. On the next page select *Saccharomyces cerevisiae* from the taxonomy browser and click on continue.

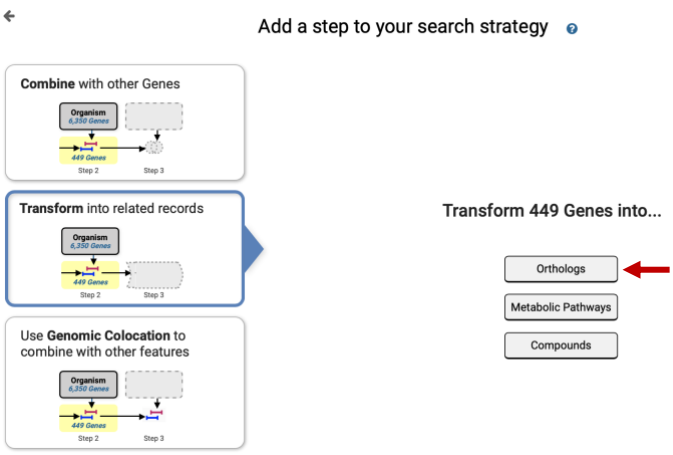


3. Configure the parameters on the next page to return each gene from step 2 whose upstream region (600bp) overlaps the exact region of a Genomic Segment in Step1 (CACGTG) and is on either strand.

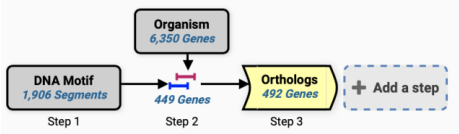


c. Identify orthologs *S. cerevisiae* genes in *Fusarium graminearum*.

All VEuPathDB sites offer tools to transform results between record types. The "Transform by Orthology" tool uses orthology clusters assigned by the OrthoMCL algorithm to enable transformation of a list of genes from one or more species to another (or more) species.



- click on add step then select the **Transform** into related records option from the left side of the popup. Next click on the *Orthologs* option.
- Select *F. graminearum* in the next popup window and click on *Run Step*.



Appendix A

Regular expression help

The following codes can be used to represent classes of amino acids.

AA property	Amino acids	Code
Acidic	**DE**	**0**
Alcohol	**ST**	**1**
Aliphatic	**ILV**	**2**
Aromatic	**FHWY**	**3**
Basic	**KRH**	**4**
Charged	**DEHKR**	**5**
Hydrophobic	**AVILMFYW**	**6**
Hydrophilic	**KRHDENQ**	**7**
Polar	**CDEHKNQRST**	**8**
Small	**ACDGNPSTV**	**9**
Tiny	**AGS**	**B**
Turnlike	**ACDEGHKNQRST**	**Z**
Any	**ACDEFGHIKLM NPQRSTVWY**	**X**

The following is a simple explanation of regular expressions.

Perl regular expressions are terms used for pattern matching in text strings, e.g. ***'aadgt', 'aa+dgt', 'a|d|c', '[mac]a'.***
Because nucleotide and amino acid sequences are text strings, regular expressions are very useful for finding motifs within sequences.
Motifs often include repetitive or ambiguous assignments at some locations. The rules and special characters used in regular expressions help define the full set of strings that match the motif pattern.
The following is a description of some of these characters and examples of how they are used.
Although regular expressions seem complicated at first, they are very useful and easy to understand after going through some examples.

Special Characters

. Match any character.

+ Matches "one or more of the preceding characters".

*****	Matches "any number of occurrences of the preceding character", including 0.
?	Matches "zero or one occurrences of the preceding character".
[ ]	Matches any character contained in the brackets.
[^]	Match any character *except* those in the brackets.
{n}	Matches when the preceding character, or character range, occurs exactly n times.
{n,}	Matches when the preceding character occurs at least n times.
{n,m}	Matches when the preceding character occurs at least n times, but no more than m times.

Here are some examples of searches.
ad+f (1 or more occurences of 'd') would match any of the following:
adf
addf
adddf
addddddf
...

ad*f (0 or more occurences of 'd') would match:
af
adf
addf
adddf
...

ad?f (0 or 1 occurence of 'd') would match:
af
adf

a[yst]c would match:
atc
asc
ayc

Specify the number of occurrences of a residue.
P{1,5} would match P from 1 to 5 times.

.{1,30} would match any amino acid 1 to 30 times so you could find a motif within 30 amino acids of something like the beginning.

Pattern Anchors

^	Match only at the beginning of the string.
$	Match only at the end of the string.

Here are examples of expressions using pattern anchors.
^mdef (e.g. a protein sequence **starting with** 'mdef') would match:

- mdef
- mdefab
- mdefaredfadfk

but not match:

- edefa
- emdefa
- eeeemdef

kdel$ (searches for proteins **ending with** 'kdel', a standard ER retention signal) would match:

- eeeekdel
- kdel

but not match :

- edefkdell
- akdeleef

Appendix B

Answers to exercise 2:

A: YXXΦ in the terminal 10 amino acids → ReEX = Y..[FTY].{0,6}$
B: YXXΦ in the terminal 20 amino acids → ReEX = Y..[FTY].{0,16}$