**Documentation for finding unidirectional best BLAST hits with reciprocal support (UBH or UBHRS)**

8/4/2016 Masayuki Onishi

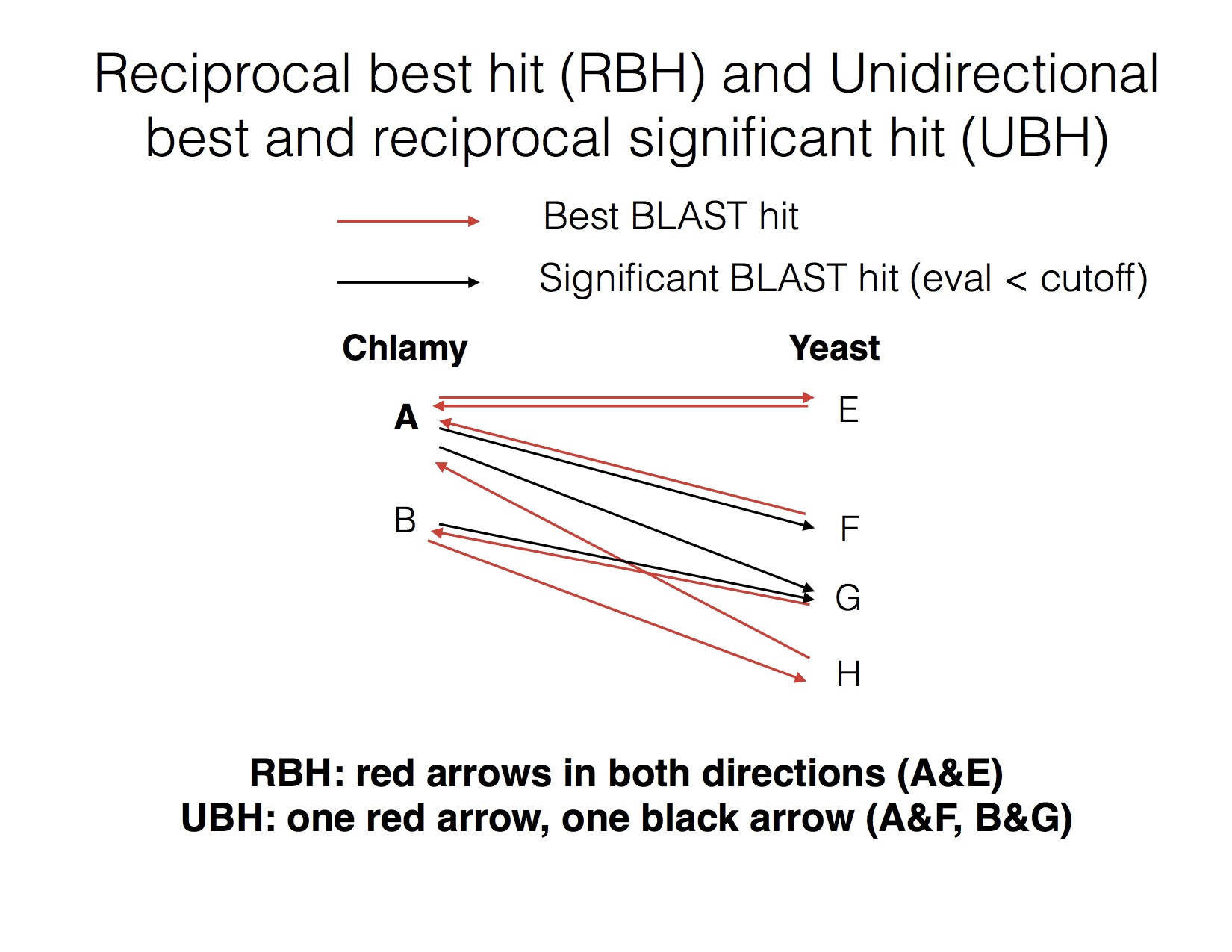
**Overview of the pipeline (detailed instruction below):**

(1) Obtain FASTA files of proteins (or transcripts) in two organisms (e.g., Chlamy and Yeast).

(2) Run genome-to-genome BLAST searches in both directions.

(3) Use script UBH.py to find UBHRS.

**The problem:**

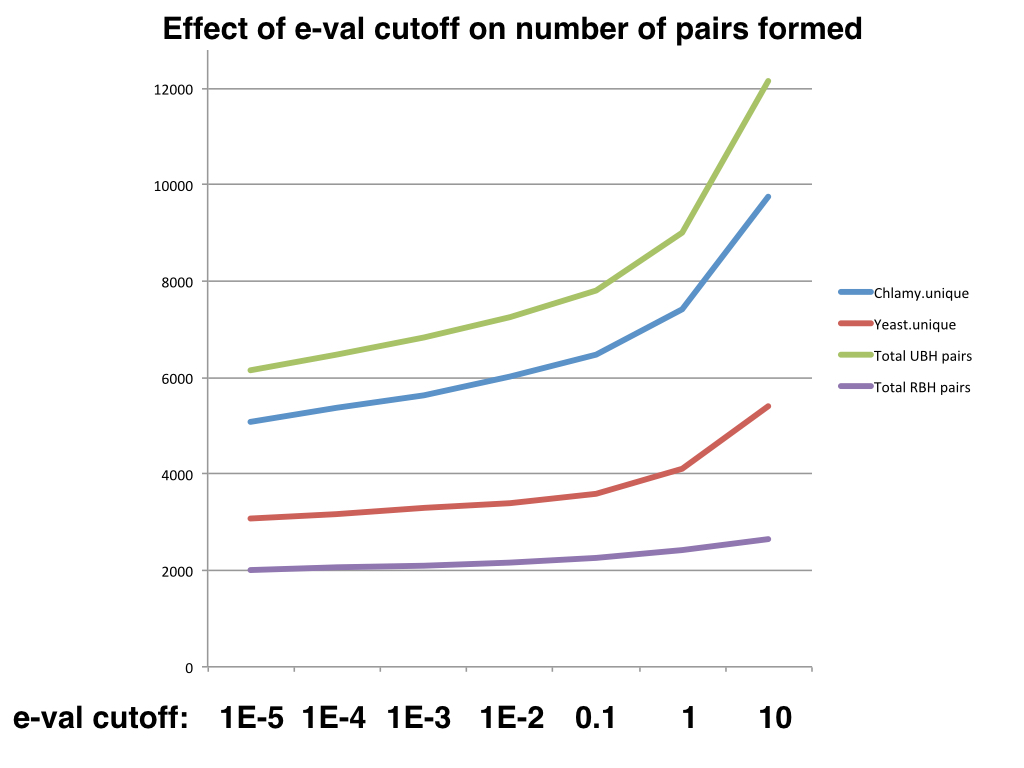


Commonly used Reciprocal Best BLAST Hit (RBH) approach ignores many genes that have been multiplicated in one genome. For example, there is only one septin (SEP1) in Chlamy, whereas yeast has seven (CDC3, CDC10, CDC11, CDC12, SPR3, SPR28, and SHS1); RBH only pairs CrSEP1 with ScCDC3. If you compare mutant phenotypes between the two organisms, Chlamy *sep1* and yeast *shs1* may share a same phenotype, but one would not be able to find that based on RBH. Similarly, if CrSEP1 and ScSPR3 (but not the other six) show a similar response to an environmental stimulus, RBH would not detect the connection.

UBHRS solves this problem by defining gene pairs that have “Best BLAST Hit in one direction, and significant (your e-value cutoff) BLAST Hit in the other.” With this method, CrSEP1 is paired with all of the seven yeast septin genes.

**There are some caveats.**

First, UBHRS is purely based on similarity, so it does not suggest an orthology. Also, it is prone to detect false-positives. For example, Chlamy actin is paired not only with yeast actin Act1 but also with actin-related proteins, and vice versa. Adjusting the e-value reduces this problem somewhat, but not much. Shown here is the numbers of pairs formed using different e-value cutoffs. I usually use e-val < 0.1, but this should be determined empirically depending on the organisms and purpose of experiments.



**Step-by-step instruction:**

(1) Obtain FASTA files of proteins or transcripts for the two organisms.

“Creinhardtii\_281\_v5.5.protein.fa” and “Yeast\_orf\_trans\_all.fasta”

(2) Run genome-vs-genome BLAST in both direction. The UBH script is written to work best with the following out format: -outfmt '10 qseqid sseqid bitscore evalue qcovs'.

E-value cutoff can be very loose here – the UBH script will apply a stricter cutoff later.

BLAST1 = “Yeast.x.Chlamy.BLASTP.e10.txt”  
 YAL001C,Cre07.g356600.t1.2,30.8,7.1,6

YAL001C,Cre05.g244701.t1.1,30.8,7.8,3

YAL001C,Cre12.g518107.t1.2,30.4,9.0,3

BLAST2 = “Chlamy.x.Yeast.BLASTP.e10.txt”  
 Cre38.g759997.t1.1,YLR153C,26.2,2.9,11

Cre38.g759997.t1.1,YDR517W,25.0,7.1,49

Cre08.g363350.t1.1,YBR056W,27.7,6.3,7

(3) Run UBH\_e.py.

Syntax is “UBH\_e.py BLAST1 BLAST2 query\_column subject\_column eval\_column eval\_cutoff UBH-list-outfile”.

To use our example, “UBH\_e.py Yeast.x.Chlamy.BLASTP.e10.txt Chlamy.x.Yeast.BLASTP.e10.txt 1 2 4 0.1 YxC.UBH.txt”

Out put file contains lines as follows.

#Unidirectional Best Blast Hits with reciprocal eval < 0.1

#A\_id B\_id eval\_A\_vs\_B eval\_B\_vs\_A BH\_dir

Cre17.g696250.t2.1 YGR162W 7e-29 1e-26 A<->B

Cre01.g009950.t1.2 YLR186W 3e-61 2e-60 A<->B

Cre13.g567450.t1.2 YPL127C 7e-06 0.0009 A<->B

Columns 1 and 2 = gene IDs, Column 3 = BLAST eval for A to B, Column 4 = BLAST eval for B to A, Column 5 = direction of best hit(s).

Note that if you select the rows with A<->B in Col 5, the resulting list will be RBH.

**Disclaimer**  
I am a beginner at Python, so forgive me if the script contains errors. Also, this UBHRS method has some caveats, so think carefully about how to use it in your study. Lastly, I have not cross-compared this method to other more complicated/sophisticated methods such as OrthoDB, InParanoid, etc.