ShortCut: short, reproducible phylogenomic cuts

Jeffrey Johnson and Krishna K. Niyogi, Howard Hughes Medial Institute, Department of Plant and Microbial Biology, University of California, Berkeley, CA 94720-3102 USA

Phylogenomic cuts are lists of genes whose distribution suggests they may be important for a trait of interest, such as virulence or drought resistance. They have historically been successful at identifying candidates for further study, but no standard methodology exists for making them or measuring their quality. They tend to involve a small number of tools—BLAST searches, set subtraction, perhaps tree clustering, and manual curation—combined in unique ways depending on the organisms and traits involved. That makes the overall process difficult to automate with a single program. ShortCut attempts to overcome that using a domain-specific language. It provides simple functions that can be rearranged to make a wide variety of cuts, as well as a novel method of measuring their robustness to changes in the search parameters. Cut scripts are reproducible, automatically parallelized, and facilitate quick comparison of many possible methods to find the most reliable list of candidate genes.

What is it?

ShortCut is a domain-specific programming language, meaning small and focused on a specific problem domain. That makes it much easier to learn than a general-purpose language like Python or R, but also limited. For example there is no way to define your own functions or read a text file. So, what can you do? You can make phylogenomic cuts! There are only a few operations to learn, but by combining them with a large number of genomes and your knowledge of the literature, you can predict which genes might be related to any biological process. You can also assess how accurate those predictions are likely to be by testing the search with known "positive control" genes.

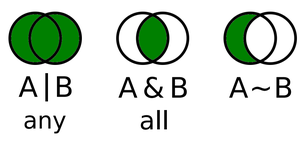
Basics

You can load genomes and convert them between several formats:

```
# load a FASTA nucleic acid file (transcriptome)
pcc6803 = load_fna "Synechocystis_PCC_6803.fna"

 # load the S. elongatus genome from two GenBank files
 # and extract the genes in FASTA amino acid format
pcc7942 = concat_fastas [gbk_to_faa (load_gbk "SynPCC7942_chr.bgk"),
                         gbk_to_faa (load_gbk "SynPCC7942_pANL.bgk")]
```

Another common task is to compare sets (lists) of various things. It works with genes and genomes, but is easier to show with numbers:

```
# starting with these lists
A = [1,2,3]
B = [2,4,6]
C = [3, 2, 1e23, 3e-9]

# ... what will each of these be? (try it!)

all [A, B, C]        B ~ (A & C)
any [A, B, C]        B ~  A & C
A | B | C
```



BLAST+

ShortCut provides the most common NCBI BLAST programs, which differ in their subject and query types. It has several variants of each function, named with suffixes.

Function	Query	Subject
blastn	nucl	nucl
blastp	prot	prot
blastx	trans	prot
tblastn	prot	trans
tblastx	trans	trans

Suffix	Meaning
`function`	"Regular" version automatically creates a database from the subject FASTA file before searching
`function_db`	Uses a prebuilt BLAST database as the subject. Useful for searching the larger NCBI databases, such as `nr` or `refseq_rna`
`function_rbh`	Does forward and reverse searches (query → subject, subject → query), and keeps only the reciprocal best hits (those where each gene is the other's top hit)
`function_each`	BLASTs the query against a list of subjects and returns a list of hit tables
`function_db_each`	Searches against a list of prebuilt databases
`function_rbh_each`	Reciprocal best hits against a list of FASTA files

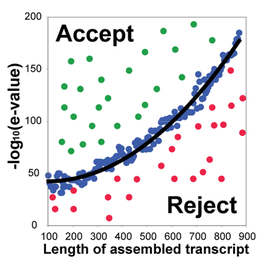
```
# blast against an NCBI database
hits = tblastn pcc7942 (blastdbget "nr") 1e-50

# find S. elongatus orthologs of Synechocystis genes
rbhs = extract_targets (blastx pcc6803 pcc7942 1e-20)
```

CRB-BLAST

Reciprocal best hits are the most common method used to find orthologs, but they can sometimes be overly conservative, missing true orthologs. For that reason, ShortCut also includes CRB-BLAST (Boursnell & Richard). For each pair of genomes, it:

1. Does a standard reciprocal BLAST search
2. Plots e-value vs sequence length of the reciprocal best hits and fits a curve to it
3. Adds non-reciprocal hits whose e-values are at least as good

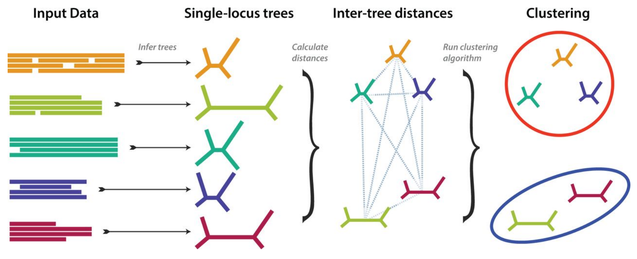
According to Aubry *et al*, this significantly improves the accuracy of ortholog assignment. Another useful feature is that it picks the e-value cutoff automatically.



Fitting a curve to determine appropriate e-value cutoffs for a pair of genomes in CRB-BLAST. From Aubry *et al.* 2014

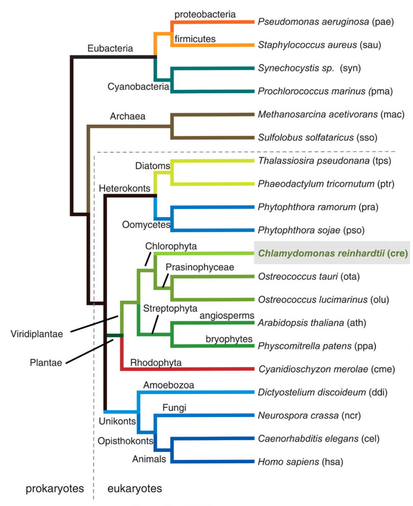
```
# make a list of chlamy genes with a reliable ortholog in at least one diatom
shared_with_diatoms = any (extract_queries_each (crb_blast_each chlamy diatoms))
```

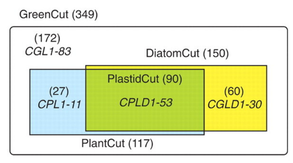
Next: cluster gene trees



After finding a large list of initial candidates using BLAST, a good way to narrow them down is by aligning each set of homologs and building gene trees from them, then clustering the trees and picking out the genes that cluster with your positive controls. TreeCl (Gori, Dezzimos *et al.* 2016) automates much of the process. Integrating it with ShortCut would improve them both, since TreeCl offers a large number of tree-building and clustering parameters that would benefit from comparison using PRS functions (see top right box).

Codifying the GreenCut





Cut	proteins in *Chlamydomonas* AND...	NOT IN...
GreenCut	*Ostreococcus*, *Arabidopsis* AND *Physcomitrella*	non-photosynthetic organisms
PlantCut	*Ostreococcus*, *Arabidopsis*, Physcomitrella AND *C. merolae*	non-photosynthetic organisms
DiatomCut	*Ostreococcus*, *Arabidopsis*, *Physcomitrella* AND a diatom	non-photosynthetic organisms

```
cyanobacteria = load_faa_each [
    "Synechocsystis_sp_PCC_6803.faa",
    "Prochlorococcus_marinus.faa"
]

other_bacteria = load_faa_each [
    "Pseudomonas_aeruginosa.faa",
    "Staphylococcus_aureus.faa",
    "Mathanosarcina_acetovorans.faa",
    "Sulfolobus_solfataricus.faa"
]

heterokonts = load_faa_each [
    "Phytopthora_ramorum.faa",
    "Phytopthora_sojae.faa"
]

# diatoms are distributed by JGI as a mapped + unmapped file each,
# which we need to combine (alternatively, could leave unmapped out)
tps = concat_fastas [
    "Thalassiosira_pseudonana_mapped.faa",
    "Thalassiosira_pseudonana_unmapped.faa"
]
ptr = concat_fastas [
    "Phaeodactylum_tricornatum_mapped.faa",
    "Phaeodactylum_tricornatum_unmapped.faa"
]
diatoms = [tps, ptr]

# separate chlamy because it's the reference,
# and use DNA rather than AA because crb_blast expects it
chlamy = load_fna "Chlamydomonas_reinhardtii.fna"

chlorophyta = load_faa_each [
    "Ostreococcus_tauri.faa",
    "Ostreococcus_lucimarinus.faa"
]

streptophyta = load_faa_each [
    "Arabidopsis_thaliana.faa",
    "Physcomitrella_patens.faa"
]

rhodophyta = load_faa_each [
    "Cyanidioschyzon_merolae.faa"
]

unikonts = load_faa_each [
    "Dictyostelium_discoideum.faa",
    "Neurospora_crassa.faa",
    "Caenorhabditis_elegans.faa",
    "Homo_sapiens.faa"
]

bacteria      = cyanobacteria | other_bacteria
viridiplantae = chlorophyta | streptophyta
plantae       = viridplantae | rhodophyta
eukaryotes    = plantae | heterokonts | unikonts

greens = plantae | cyanobacteria
others = unkonts | heterokonts | other_bacteria

green_hits  = extract_queries_each (crb_blast_each chlamy greens)
other_hits  = extract_queries_each (crb_blast_each chlamy others)
plant_hits  = extract_queries_each (crb_blast_each chlamy plantae)
diatom_hits = extract_queries_each (crb_blast_each chlamy diatoms)

greencut   = all green_hits ~ any other_hits
plantcut   = all plant_hits ~ any other_hits
diatomcut  = plantcut & any diatom_hits
plastidcut = plantcut & diatomcut
```

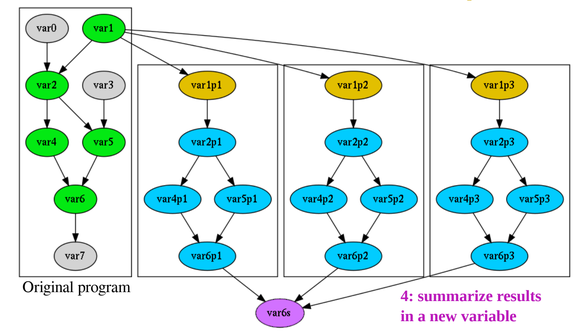
Left: Merchant *et al.* 2007. Evolutionary relationships of 20 species with sequenced genomes used for the comparative analyses in [the GreenCut] include cyanobacteria and nonphotosynthetic eubacteria, Archaea and eukaryotes from the oomycetes, diatoms, rhodophytes, plants, amoebae and opisthokonts. The GreenCut comprises 349 Chlamydomonas proteins with homologs in representatives of the green lineage of the Plantae (Chlamydomonas, Physcomitrella, and Ostreococcus tauri and O. lucimarinus), but not in nonphotosynthetic organisms. Genes encoding proteins of unknown function that were not previously annotated were given names on the basis of their occurrence in various cuts. CGL refers to conserved only in the green lineage. The GreenCut protein families, which also include members from the red alga Cyanidioschyzon within the Plantae, were assigned to the PlantCut (blue plus green rectangles). CPL refers to conserved in the Plantae. GreenCut proteins also present in at least one diatom (Thalassiosira and Phaeodactylum) were assigned to the DiatomCut (yellow plus green rectangle). CGLD refers to conserved in the green lineage and diatoms. Proteins present in all of the eukaryotic plastid-containing organisms in this analysis were assigned to the PlastidCut (green rectangle). CPLD refers to conserved in the Plantae and diatoms. The criteria used for the groupings associated with the GreenCut are given in the lower table.

Right: Roughly equivalent ShortCut script. A proteome (FASTA Amino Acid file) is loaded for each species. They are grouped into lists representing the most specific taxa, with higher taxa represented by unions of those lists. Two more groups are defined for convenience: "green" and "non-green" species. CRB-BLAST searches are done between *Chlamydomonas* and each of the other genomes (this differs from the simple BLAST with cutoffs used previously). Finally, the "cuts" are made by extracting the list of *Chlamydomonas* genes with putative orthologs in each species, and comparing them as sets. Note that even though there are many duplicate BLAST searches specified—for example all the species in plantae are also in greens—caching ensures each operation is only done once. It is also possible to calculate one cut at a time, skipping any steps whose results aren't used.

Permute, Repeat, Summarize

Making a cut involves choices: which genomes to include, how much to trust their gene annotations, which BLAST functions to use, which e-value cutoffs to apply at each step... How can you be sure the parameters you picked are reasonable? ShortCut has a novel solution: duplicate parts of the program, re-run them starting from alternate values, and see what changes.

1: Find variables that need to be recalculated **2: Duplicate them once for each permutation** **3: Replace var1 with each of its permutations**




```
var6s = repeat_each var6 var1 [var1p1, var1p2, var1p3]
```

Visualizing the PRS pattern. You have the program on the left and want to know, "What happens to var6 if I change var1?". The repeat_each function recalculates var6 starting from 3 alternate versions of var1. This example is mostly "repeat" with trivial "permute" and "summarize" steps. A more practical example might look like this:

```
# blast S. elongatus genes against Synechocystis with a standard cutoff
cutoff = 1e-10
hits = extract_queries (blastp pcc7942 pcc6803 cutoff)

# re-run it with a range of cutoffs and report the number of hits for each
cutoffs = [1e-5, 1e-10, 1e-20, 1e-50, 1e-100, 0]
lengths = repeat_each (length hits) cutoff cutoffs
```

Next: cross-validation

This methodology can be extended to automatically optimize parameters. For example you could try a range of e-values and pick the one the maximizes the number of known genes rediscovered while minimizing the number of total candidates. It is also considered good practice when training an algorithm to hold some of the test data in reserve for measuring performance at the end. This guards against over-fitting (optimizing the algorithm for your exact test data, only to find later that other data are different). It could be done fairly easily in ShortCut by splitting a list of known positive-control genes into training and validation lists, optimizing to discover genes in the training list, and finally reporting the number of validation genes rediscovered.

Live demo!

Try any of the code on the poster, or play around on your own. If some function crashes or you get stuck, press **Ctrl-C** to reset the demo. After all, this is a work in progress...

Howard Hughes Medical Institute

PLANT & MICROBIAL BIOLOGY
COLLEGE OF NATURAL RESOURCES
UNIVERSITY OF CALIFORNIA, BERKELEY

References & Acknowledgements

Aubry, S., Kelly, S., Kümpers, B.M.C., Smith-Unna, R.D., and Hibberd, J.M. (2014). Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. PLOS Genetics 10, e1004365.

Boursnell, C., and Richard, S.-U. (2017). crb-blast: Conditional Reciprocal Best Blast.

Gori, K., Suchan, T., Alvarez, N., Goldman, N., and Dessimoz, C. (2016). Clustering Genes of Common Evolutionary History. Mol Biol Evol msw038.

Merchant, S.S., Prochnik, S.E., Vallon, O., Harris, E.H., Karpowicz, S.J., Witman, G.B., Terry, A., Salamov, A., Fritz-Laylin, L.K., Marechal-Drouard, L., *et al.* (2007). The Chlamydomonas Genome Reveals the Evolution of Key Animal and Plant Functions. Science 318, 245–250.