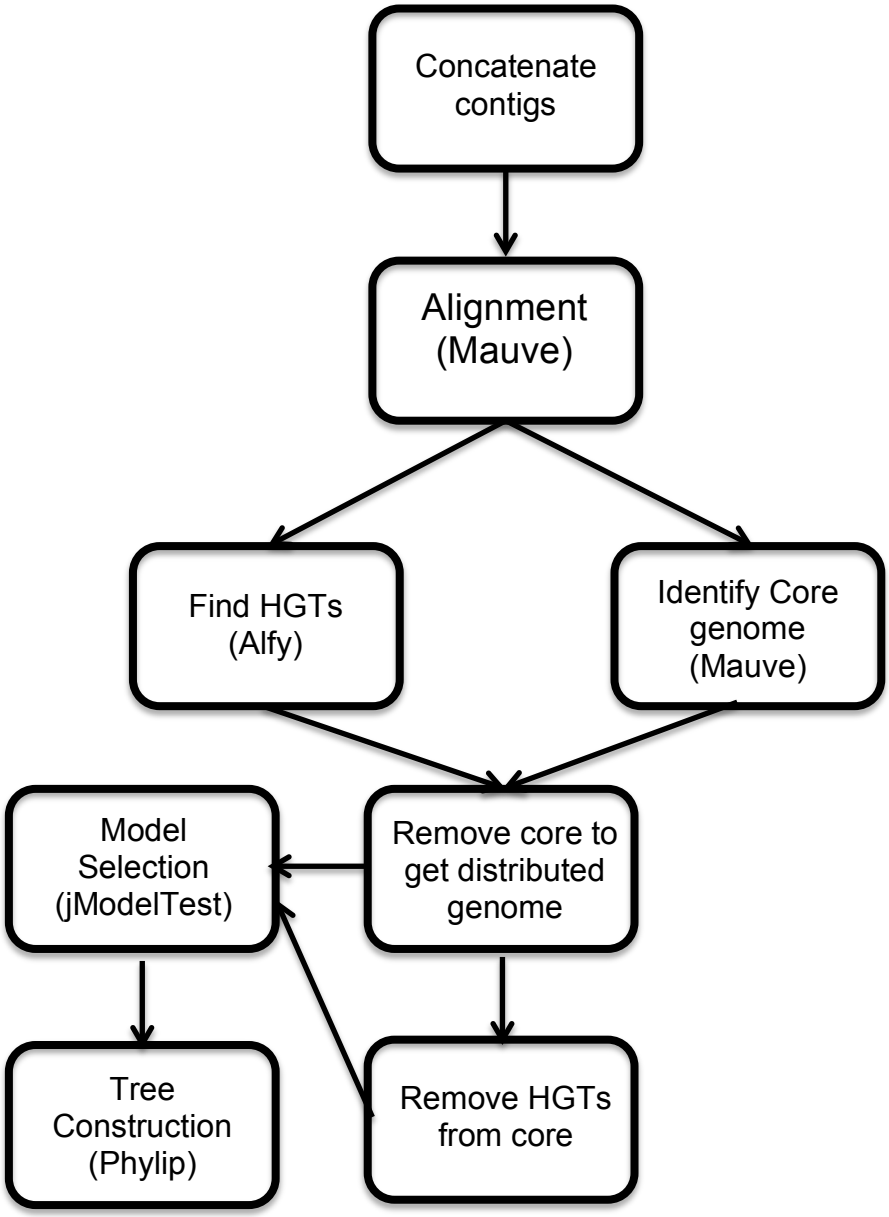
Pipeline for Analysis of Horizontal Gene Transfer in Bacterial Genomes

Stuti Agrawal
Rebecca Elyanow
Luigi Leung
Prateek Tandon
Yiming Xin

Contents


Concatenate contigs
Alignment (Mauve)
Find HGTs (Alfy)
Identify Core genome (Mauve)
Model Selection (jModelTest)
Remove core to get distributed genome
Tree Construction (Phylip)
Remove HGTs from core

1 Biological Problem

Whole genome sequencing is become more and more available as sequencing tehcnology rapidly advances. With a growing amount of whole genomes becoming available, the task of analyzing these genomes becomes increasingly important.

In bacterial genomes, even very closely related strains can have significant genetic variation due to genomic recombination. Through horizontal gene transfer, bacteria have to ability to transfer genes between neighboring cells, leading to very rapid evolution of the genome. Often, genes that have undergone horizontal transfer are important for the survival and reproduction of the bacterial cells. As such, these regions of recombination can unveil important features about a community of related bacterial strains, such as what genes cause vaccine susceptibility, antibiotic resistance, and pathogenicity. In order to get a complete understanding of a community of bacteria, it is important to study both the vertical descent as well as the lateral transfer of genes from strain to strain.

2 Pipeline Overview

The purpose of the pipeline is to take in as input a set of whole genome sequence data, identify the core and distributed genomes as well as regions of horizontal gene transfer, and create phylogenetic trees for the regions of horizontal gene transfer. This is accomplished using four publicly available software packages: MAUVE, Alfy, jModelTest, and Phylip.

Mauve is used to align the sequences as well as to identify the conserved regions among all sequences (after removing HGTs, this makes up the core genome). Alfy is used to identify regions of horizontal gene transfer and input them into a dictionary. For each entry in the dictionary, a phylogenetic tree is constructed. The most appropriate model is chosen using jModelTest and the tree is created using Phylip.

The pipeline can be summarized as follows:

1. Concatenate contigs in each genome (these files will be used for the rest of the pipeline)
2. Align genomes using Mauve and store the .backbone file (containing start and end positions of the conserved regions amongst all genomes)
3. for each genome in the set, run Alfy with the selected genome as a reference and every other genome as a comparison. Store the resulting homology regions in a dictionary. If the results are determined significant by kr, then they will be labeled as regions of horizontal gene transfer
4. Identify the core genome based on the .backbone file produced by Mauve and extract HGT regions identified by Alfy, create a phylogenetic tree of the core genome (using

jModelTest and Phylip)

5. Remove the core to get the distributed genome and create a phylogenetic tree of the core genome
6. For each HGT region create a phylogenetic tree (using jModelTest and Phylip)

2.1 GUI

The GUI is a web based environment developed in using Python CGI. We choose to use a web based environment so that our pipeline could be utilized on any computer. It is hosted via an Apache web server locally and on CMU's AFS server remotely. The GUI allows the user to enter the data they wish to analyze and will send the user an email when the job is finished that allows them to view the results. The results are presented as images of phylogenetic trees as well as .fasta files of the core and distributed genomes.

3 About Mauve

Mauve was used for alignment because it was designed to deal specifically with genomes that may contain regions that are not the result of pure vertical descent. The creators of Mauve write in their paper, "We present methods for identification and alignment of conserved genomic DNA in the presence of rearrangements and horizontal transfer" [1]. The Mauve algorithm can be described as follows [1]:

1. Find local alignments using Multiple Maximal Unique Matches (multi-MUMS)
2. User multi-MUMs to calculate a phylogenetic guide tree
3. Select a subset of multi-MUMs to use as anchors (which are partitioned into Locally Collinear Blocks (LCBs)
4. Perform recursive anchoring to identify additional alignment anchors outside/within each LCB
5. Perform progressive alignment of each LCB using guide tree

This results in a global alignment of each locally collinear block that contains sequence elements which are conserved among all of the analyzed genomes. Information about these conserved sequence elements can be found in the .backbone output file, which is used in our pipeline for identification of the core genome (regions of the genome that are conserved amongst all strains under study).

3.1 Testing

The creators of Mauve tested their program against other commonly used alignment tols, like Multi-LAGAN and Shuffle-LAGAN. According to their analysis, Multi-LAGAN aligns more divergent genomes better than Mauve, but for closely related sequences that have undergone modest amounts of substitution and inversion, Mauve performs better than both Multi-LAGAN and Shuffle-Lagan [1].

Mauve was also tested with user generated genomes of about 2000kb (about the size of a bacterial genome) that contained regions that were conserved amongst all genomes. Mauve correctly identified the regions that were conserved amongst all genomes (the core genome.)

4 About Alfy

Alfy was designed to detect horizontal gene transfer in bacterial genomes [4]. This is exactly our intended use of the program. Alfy is able to detect regions of horizontal gene transfer based on the lengths of exact matches between pairs of sequences (with long matches indicating close homology and short matches indicating distant or no homology) [4].

We Alfy in our pipeline to detect HGT events. If we want to identify the HGT regions of a genome(say seq1), we run the Alfy with seq1 as query and all the other strains genome as subjects. Alfy will output all the potential HGT position intervals with the strains where seq1 might gets HGT from. We run Alfy for all sequences of interest in this way. Then we use kr [11], another software package working as an estimator of the pairwise number of substitutions between long DNA sequences, to generate a correlation matrix of all sequences. After that, we form a closest mates group for all the sequences we test by choosing the corresponding sequences in the matrix with a value below the threshold (set to 0.05). Then we filter the output of Alfy by removing the regions whose subject sequences are in the closest group of query sequence.

4.1 Testing

The creators of Alfy tested the program on HIV-1 recombinant strains and E.coli genomes. According to their tests, Alfy could classify recombinant group M HIV-1 strains as accurately and reliably as the commonly used NCBI tool. Alfy was also tested for identifiation of horizontal gene transfer in E.coli genomes, which is essentially equivalent to our problem.

The accuracy of this method seems acceptable given the result reported in the paper[4],[11]. However, in the process of testing our pipeline, we found that some sequences have no closest-mates group output, which means the sequence is not close to any of other sequences

according to the matrix we generate. In this situation, the whole sequence of such strain would be identified as HGT regions, which is an incorrect conclusion. We propose to avoid this pitfall by choosing sequences which we can assume have relatively high homology as inputs.

5 About PHYLIP

PHYLIP is used to create phylogenetic trees for the core genome and each HGT region. First, the best model is chosen using jModelTest (which takes as in put a sequence and outputs a ranked list of possible models). PHYLIP is then run on using this model. It can implement either parsimony based trees (DNAPARS) and maximum likelyhood trees (DNAML) as chosen by the user. Before estimating phylogenies, it performs 100 bootstraps using SEQBOOT. It can also compute consensus trees (CONSENSE).

6 Comparison to other software

Through our research, we have not found another piece of software that takes in raw sequence data and performs whole genome alignment, HGT identification, tree building, and annotation. There are many programs that solve pieces of the problem (some of which we have implemented in our solution), but none that solve every part of our problem.

For our comparison, we will explore available software packages that offer solutions to parts of our pipeline: whole genome alignment, HGT identification, tree building, and annotation.

6.1 Whole Genome Alignment

There are many software packages that deal with whole genome alignment including: BLAST [8], MUMMER [6], LAST [7], and Mugsy [5].

We choose to use MAUVE [1] over these other software for a few reasons:

1. It runs from the command line on Mac, Linux, and Windows
2. It does not require a reference sequence
3. It can take both genbank and fasta files as input
4. It outputs a .backbone file which can be utilized for identification of the core genome

6.2 HGT identification

We looked into two HGT identification packages, RDP and Clonal Frame, before settling on the package we are currently using, Alfy. We decided not to use RDP [9] because 1. it only runs on Windows machines and 2. The current version is still unstable.We decided not to use Clonal Frame [10] because 1. it only runs on Mac/Linux machines and 2. it has a history of being very slow and difficult to use (according to discussion with Luisa Hiller). We choose Alfy [4] because it can run on all platforms and, testing with 15 genomes, it runs in under 5 minutes.

6.3 Tree building

Numerous tree construction algorithms exist including: MEGA, MrBayes, and PAUP. We chose to use PHYLIP [3] because it is a geneal-purpose tree building package that is well documented, can run on all operating systems, and is easy to implement from the command line.

7 Setbacks and future additions

During the process of developing the pipeline we ran into several roadblocks including:

1. Finding software that could efficiently and accurately determine HGT regions
   We tried implementing RDP and Clonal Frame with no success before finding and settling on Alfy
2. Creation of the web interface using Django
   No of the team members had any prior experience with Django. It turned out that the learning curve was too steep to be able to create a successful web interface in the given time, so we ended up switching to Python CGI
3. Visualization of HGT trees
   Setbacks in the rest of the pipeline prevented us from focusing on visualization. We are currently working on how to implment fluid visualization of many trees in our GUI.
4. Annotation of HGT regions
   Setbacks in the rest of the pipeline prevented us from working on implementing annotation of HGT regions. We hope to imlement this feature in the future

References

[1] Aaron E. Darling, Bob Mau, and Nicole T. Perna. 2010. progressiveMauve: Multiple Genome Alignment with Gene Gain, Loss, and Rearrangement. PLoS One. 5(6):e11147.

[2] Darriba D, Taboada GL, Doallo R, Posada D. 2012. jModelTest 2: more models, new heuristics and parallel computing. Nature Methods 9(8), 772.
Guindon S and Gascuel O (2003). A simple, fast and accurate method to estimate large phylogenies by maximum-likelihood". Systematic Biology 52: 696-704.

[3] Felsenstein, J. 1989. PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics 5: 164-166.

[4] Domazet-Lošo M, Haubold B. Alignment-free detection of local similarity among viral and bacterial genomes. Bioinformatics. 2011;27:1466–1472. doi: 10.1093/bioinformatics/btr176.

[5] Angiuoli SV and Salzberg SL. Mugsy: Fast multiple alignment of closely related whole genomes. Bioinformatics 2011 27(3):334-4

[6] "Versatile and open software for comparing large genomes." S. Kurtz, A. Phillippy, A.L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S.L. Salzberg, Genome Biology (2004), 5:R12.

[7] Adaptive seeds tame genomic sequence comparison. SM Kielbasa, R Wan, K Sato, P Horton, MC Frith, Genome Research 2011.

[8] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

[9] Martin DP, Lemey P, Lott M, Moulton V, Posada D, Lefeuvre P. (2010). RDP3: a flexible and fast computer program for analyzing recombination. Bioinformatics 26, 2462-2463.

[10] Inference of Bacterial Mi-croevolution Using Multilocus Sequence Data" by X. Didelot and D. Falush, Genetics, Vol. 175, 1251-1266, March 2007.

[11] Domazet-Lošo M, Haubold B. Alignment-free detection of horizontal gene transfer between closely related bacterial genomes. Mobile Genetic Elements 1:3, 230–235; September/October 2011;G2011 Landes Bioscience.