Notes on HLA Typing Algorithms

Jean-Baptiste Reynier
MPCS 53112, University of Chicago

December 14, 2019

seq2HLA:

seq2HLA is a HLA typing algorithm that takes as input RNA-seq data. It first maps the RNA-seq sequences to the alleles of the IMGT-HLA database using an efficient mapping algorithm called Bowtie. Bowtie is an improvement upon the simple pairwise alignment algorithms, in that it uses the Burrows–Wheeler transform to speedup the alignment process: this method rearranges nucleotides in each sequence into runs of similar characters, which allows for compression, and reduces memory usage. It then indexes the sequences in the genome to map each input sequence to the genome sequences more rapidly.

After aligning the RNA-seq reads to the reference genome, the seq2HLA algorithm counts the number matches to sequences for each HLA allele, and finds for each gene the allele with the highest count. It then removes the reads associated with those alleles and runs the algorithm again to find the second allele for each gene. In order to differentiate between homozygous (same allele for a gene) and heterozygous (two different alleles for a gene) cases, seq2HLA looks at the ratio of the number of sequences matched for each allele: if the second allele is above the zygosity threshold, both alleles are kept, but if the ratio is below the threshold, the individual is declared homozygous for that gene.

arcasHLA:

arcasHLA also allows only RNA-seq sequences as input for the algorithm, and these sequences have already been aligned to the reference genome. The algorithm selects the RNA-seq reads mapped to the chromosome 6 (which is where all the HLA genes are located). It then uses a de Bruijn graph to realign those sequences.

This directional graph has nodes which represent k-1 length subsequences, and each edge in the graph is an additional nucleotide linking a subsequence to a left-shifted subsequence of length k-1. Removing the last nucleotide on the left of a subsequence and adding the nucleotide of the edge should produce the subsequence the edge is pointing to (e.g. subsequence ACTAC is linked to CTACA with edge A). Each subsequence in the reference is hashed to a reference index, which allows the mapping algorithm to avoid a base by base alignment.

arcasHLA creates a graph with the alleles of the IMGT database, and then hashes the RNA-seq reads from the individual to the graph index. It then looks at the intersection of the subsequences that each read hashes to, to see which alleles the subsequences align to. The algorithm then counts the number of reads per group of alleles with matching path, and assigns for each allele a number of counts: this count is weighted by the frequency of that allele within the population of the individual (e.g. African, European etc.), and normalized by read length to produce a relative abundance score. It then implements an iterative procedure to maximize likelihood function while culling out unlikely alleles: counts are recalculated according to previous relative abundance, and relative abundance is then recalculated using updated counts (again taking into account read length). Those alleles with relative abundance below 0.1 are removed. If after convergence, there are still multiple chosen alleles, the ones with the overall most reads are chosen, and arcasHLA checks for homozygosity by looking at the relative abundance ratio between the two chosen alleles.

HLA-HD:

HLA-HD creates a dictionary of all introns/exons in the HLA IMGT database, and append series of "N" nucleotides to the ends of all sequences (the nucleotides "N" represent any nucleotide A, C, T or G). It then maps reads of the individual to the dictionary with no mismatches allowed in exons, and up to two mismatches allowed in introns. If a read maps to an exon ($\geq 50\%$), or if a read maps to an intron ($\geq 50\%$) in the database, the read is kept (that is, the read can have up to 50% of alignment to the added "N" nucleotides at the end of the reference sequences). The algorithm then calculates the sum of all matches for an allele, where the weight for each match takes into account the number of matches of the read to alleles of the same exon/intron, the number of matches to other introns/exons of the same gene, and the number of matches to other genes. It also normalizes the count by the length of the read. It then looks over all possible allele pairs (e.g. A and B), and for each allele pair, it sums the weighted match count of A&B, A&(!B) and (!A)&B. It selects the best pair out of all possible. Again, it checks for homozygosity looking

at the ratio match count of allele A to B.

HLA-LA:

HLA-LA creates a multiple alignment of all alleles of each gene in the HLA IMGT database. A population graph is then created, representing all possible sequences for that gene: the aligned sequences of the alleles are merged in the regions of exact sequence match. Therefore, the paths in the graph are identical to the original sequences of each allele, except for the possible introduction of gaps during the alignment. To match an individual's sequence reads to the graph, each read is first mapped to a reference sequence for each graph. Once it is mapped to the reference sequence, it can easily be projected onto the graph itself, finding the best possible path through the graph. After further optimization of the alignment, the likelihood scores of each possible allele pair is calculated: this is the product of the sum of the match score of each allele in the pair, for each sequence read mapping to that gene region, that is:

$$\prod_{reads\ mapping\ to\ region} \left(\frac{1}{2} * score_{allele1} + \frac{1}{2} * score_{allele2} \right)$$

This allele/read match score has a reward for exact nucleotide matches, as well as costs for insertions and deletions. The pair with the highest likelihood score is selected.

Kourami:

The Kourami program takes as input aligned sequences to the reference genome, and extracts all reads which map to known HLA genes in chromosome 6. Similarly to HLA-LA, Kourami creates a graph for each gene, using the multiple sequence alignments of all alleles in the HLA IMGT database. The biggest difference with the HLA-LA method is that Kourami allows for the HLA allele graph to be modified: if an individual's reads do not map perfectly to the reference graph, the graph is updated (in case of a deletion, insertion or mismatch). This therefore allows the Kourami algorithm to find novel HLA alleles that are not yet in the IMGT database. To find the best allele pair, the algorithm focuses on the parts of the HLA graph where variations are captured, and ignores all the sequences identical to all alleles (thus transforming the graph into a "bubble graph", where each bubble represents variation between alleles). The algorithm's aim is to find the two best paths for each bubble (these paths can be the same in the case of homozygous individuals):

Kourami iterates through all possible pairs of bubble paths and calculates the posterior probability of each pair (using the number of reads aligned to the bubble). It then computes the conditional probability, and finds the best pair of bubble paths for each gene. Finally, the algorithm searches the list of alleles to find the ones that are the most consistent with the best bubble paths found earlier.