# The EuPathDB High Speed SNP Search (HSSS) system

This documentation describes the High Speed SNP Search (HSSS) system developed by Steve Fischer at the EuPathDB project. The searches described here typically process 100 or more resequenced parasite genomes in under 10 seconds. For live examples, see the SNP searches on PlasmoDB or ToxoDB.

Contact: sfischer@upenn.edu

## == The Data underlying the searches ==

The fundamental input to these searches is a set of resequenced genomes, here referred to as *strains*, that come from the same species. They have been sequenced in short reads. The reads are mapped to a reference genome, where they tile across it. The height of the reads at a given location is called the "coverage." If the coverage is less than five we ignore the reads at that location. If there are more than five reads at a location, then we look at the largest number of those reads that agree with each other. The percentage that these reads are of all the strain's reads at that location is called the "read frequency." In the searches described below, the user will provide a threshold for this. Locations with reads that do not have an allele at or above that threshold, or those that have no reads at all, are considered "unknown" and are ignored.

All this information for all the strains (at a given read frequency cutoff) is collated so that we have at each location of the reference genome, for every strain, one of the following:

* a single allele that agrees with the reference genome
* a single allele that differs from the reference genome
* more than one allele, where one agrees with the reference genome
* more than one allele, where all differ from the reference genome
* unknown

Locations that have at least strain that has at least one allele that differs from the reference genome are considered a "global SNP location." That is to say, this is a location where, when considering all available strains, there is a known SNP. (If one were to consider only a subset of the strains, that location might not have a SNP. That is what the searches do.)

**Definition:** A *global SNP* is a location on the genome in which, when considering all available strains, there is at least one strain with an allele that is different than the reference.

**Definition:** A *local SNP* is a location on the genome in which, when considering a subset of strains (those of interest), there is at least one strain with an allele that is different than the reference. A local SNP must also be a global SNP.

## == Search basics ==

The HSSS system supports a number of searches. They divide into two types: (1) given one group of strains (of a single species) and some parameters, find (local) SNPs among that group; (2) given two groups of strains (of a single species) find SNPs that differentiate the groups. The first of those comes in different varieties based on some filters applied to resulting SNPS (genomic location or gene location).

The input to the searches are files that describe the alleles of each strain for each global SNP location. For each strain there are a number of files, each at a predetermined read frequency cutoff (for example 20%, 40%, 60% or 80%). We use this quantization for performance reasons. The user chooses the input strains (either one set or two sets, depending on the search). The C code processes the selected input files, merging pairs of files in parallel, and passing those to a next parallel tier of merging. Because the files are sorted by SNP location, merging is efficient. At the end of the merging the output has information for all involved strains in one stream, collated into SNP locations. In the one group searches this stream is passed to filters to find SNPs that have the desired level of polymorphism, as well as other filters like gene locations. In the two group search each group is reduced to one of those streams. Each of those is passed to a program that collapses each to a consensus allele for each SNP. Then those to consensi are compared to find SNPs between them.

All processing is achieved with UNIX streams. The only disk i/o is reading the initial input files and writing the final result. This is key to the speed of the system. Most steps in the processing take in two streams and emit one. stdin does not work for that, as it is only a single stream. Instead we use UNIX FIFOs. These are simply named streams that appear to the consuming program like a file handle.

The structure of stream processing is a reducing tree of processes. The size of the tree is determined by the number of strains the user chooses as input. The search system implements this tree by dynamically generating a shell script that is custom for a particular search. The script passes arguments into the programs it calls that reflect the user's choice of search parameters. For parallelism, the script puts all the processes in the background. They are tied together by the FIFOs the consume and produce.

The programs that do the most intensive data processing, for example the merging of strain files, is coded in C for speed. The final filtering is sometimes done in Perl.

The strain files are encoded in the most economical fashion, to keep the very large files as small as possible. This reduces I/O which is the main expense in processing this kind of data. The files are binary. Strain names and sequence source\_ids are encoded as simple integer IDs (a mapping of these IDs are stored in separate files).

The ultimate mapping of SNPs found by an HSSS search to SNP ID is based on the computability of SNP ID. A SNP ID is constructed as follows: NGS. + sequence \_id + location.

## == Data files ==

The workflow creates a set of data files for use by HSSS. They are stored in the dedicated directory and are organized into a directory structure something like this:

PlasmoDB/

Pfalciparum3D7/

readFrequency20/

readFrequency40/

readFrequency60/

readFrequency80/

PvivaxBlah/

…

The readFrequency directories contain the data files. The files are created with a particular read frequency cutoff as per the directory name. The files in these directories are:

**(1) strain files.** Each file represents the allele(s) of single strain at a particular read frequency threshold, at each location in the genome that has any variation at that location at that read frequency. (Alleles that are not represented by that percentage of the reads at that location are excluded). The file is named with the "HSSS strain ID", which an integer that has been mapped to the strain name. The files describe the alleles of a given strain for each SNP location. They are binary. The rows are sorted by seq\_id, location. It is permitted for a SNP to have more than one row, which means that this strain had more than one allele at that location at this read frequency threshold. This can only happen at read frequency thresholds <= 50%, and represents either heterozygosity (diploid organisms), unpure strains or sequencing errors.

The files are binary with 4 columns:

* seq\_id. 2 bytes. the HSSS seq ide, an integer mapped to the sequence source id. (They start at 1.)
* location. 4 bytes. the location on the sequence of SNP
* allele. 1 byte. 1=A, 2=C, 3=G, 4=T, 0=unknown (alleles with other IUPAC codes, eg, Y, are set to 0, ie, ignored)
* product. 1 byte. the ascii encoding of the upper case product, eg, A=65. \*=42 (stop codon) 0=non-coding.

If an allele is "like reference," ie, the same as the reference genome (that genome to which all the strain sequences were mapped), then that allele is not included in the strain file. This is a form of compression. (The gamble is that there are fewer unknown locations than like-reference, and that is increasingly true as there are more strains causing more SNP locations, given that coverage is usually pretty good in NGS resequencing). If a strain has more than one allele at a location then both are included in the file even if one is like-reference. If the reference has an IUPAC code for an ambiguous base and the strain has that same code, then the strain is considered like-reference, and no row is included in the file.

**(2) reference genome file.** Same columns as strain files. This file describes the alleles in the reference genome. (The genome to which all the strains were aligned.) Every known SNP for the organism has a row in this file. The only alleles with value 0, ie, unknown, are those where the allele in the genome is an IUPAC ambiguous base pair code.

**(3) strain IDs file.** Maps integer strain IDs to strain names. The IDs start with 1.

**(4) sequence IDs file.** Maps integer sequence IDs to sequence source\_ids. The IDs start with 1.

## == Runtime streams ==

These streams are produced as an HSSS search executes.

**(1) merged strain streams.** Produced by hsssMergeStrains. These files represent the merging of 0, 1 or more strain files. Like strain files they are binary and sorted by seq\_id and location. For a particular SNP, all alleles from a given strain are in contiguous rows but otherwise the strains are not ordered. The files have five columns. The first four are the same as in a strain file; the fifth is the strain\_id (2 bytes). These files introduce a form of compression not used in the strain files. If a given SNP has more than one unknown, the multiple unknowns may be compressed into a single row. That row will have a -1 in the product column and a count of unknowns in the strain\_id column. A SNP may have 0, 1 or more than 1 such compressed rows.

**(2) consensus streams.**  Produced by hsssMergedToConsensus. These files represent the consensus of a set of strains. For each SNP, the set of alleles among all the strains at that SNP is reduced to a major and minor allele, where the former is the most popular and the latter the second most popular. If there are ties the choices are arbitrary. A SNP is not reported in this file if not enough strains have reads at that location (as specified by the unknowns threshold parameter to hsssMergedToConsensus). The columns are:

* seq\_id. same as in strain files
* location. same as in strain files
* major\_allele. 1 byte. same encoding as strains file (except never 0).
* major\_allele\_product. 1 byte. same encoding as strains file.
* major\_allele\_product\_is\_variable. 1 byte. 0=all major alleles have the same product. 1=they do not.
* minor\_allele.
* minor\_allele\_product.
* minor\_allele\_product\_is\_variable.
* major\_allele\_per\_ten\_thousand. 2 bytes. percent of alleles that are the major allele, times 100 for precision
* minor\_allele\_per\_ten\_thousand.
* is\_triallelic. 1 byte. 1=is triallelic, ie, has more than two alleles.

**(3) SNPs streams (one group searches).** Produced by hssFindSnpsInOneSet. Represents the SNPs found in a single group search. It is tab delimited text, with these columns:

* seq\_id. same as in strain files
* location. same as in strain files
* knownPercent. a float, to one decimal point. The % of strains having reads at this location.
* polymorphismsPercent. a float, to one decimal point. The % of known alleles that are not the major allele.
* productClass.
* 0=noncoding (no allele has a product or stop codon)
* 1=synonymous (all products agree)
* 2=nonsynonymous (not all products agree)
* negative=nonsense (at least one allele has a nonsense product)

**(4) SNPs streams (two groups search).** Produced by hsssFindSnpsInConsensii. Represents the SNPs found across two groups of sequences. Each group (A and B) has been reduced to a consensus. This output shows details of each of those consensuses at SNP locations where they differ from each other. Tab delimited text, with these columns:

* group A seq id. HSSS seq ID.
* group A location.
* group A major allele.
* group A major allele percent. The percent of alleles in the consensus that are the major allele.
* group A is triallelic.
* group A major product.
* group A major product is variable.
* same columns for group B

## == Final output files ==

The final output of a search is written to a directory accessed by the WSF plugin. The plugin passes the rows to the WSF framework, unmodified. The final output files contain the values the user sees. A given search has one of these three types of output file:

**(1) SNPs found among a single set of strains.** This is output by searches that take as input a single set of strain files. Some of these searches have filters that remove SNPs from the stream (eg, by genomic location), and this output is after those filters. This file has the same information as the SNPs stream (one group searches), but in a user-readable format. The SNP source ID is formed from the raw seq\_id and location columns, and the productClass is converted from a number code to words.

**(2) SNPs found among two sets of strains**. This is output by the search that takes as input two sets of strain files. This file has the same information as the SNPs stream (two group searches), but in a user-readable format. The SNP source ID is formed from the raw seq\_id and location columns, and the productClass is converted from a number code to words.

**(3) Genes that have SNPs found among a single set of strains.** This is produced by analyzing a stream of SNPs (single set of strains), looking at the SNPs within gene boundaries, and outputting the genes whose SNPs meet the specified criteria, as well as summary statistics about those genes' SNPs. Tab delimited text:

* gene id
* SNP density
* ratio non-syn/syn
* syn count
* non-syn count
* non coding count
* nonsense count
* SNPs count

## == Types of searches ==

There are two types of searches: those that process a single set of strains and those that process two sets of strains.

In the first case the user selects a set of strains and a read frequency threshold. That determines a set of input files (one per strain at that read frequency). These files are merged (in parallel) to form a single stream containing the collated information for each global SNP location. That stream is processed to identify either SNPs or genes.

Here is an illustration, where the user has selected four strains. (The tree would expand to the left if they had selected more). The numerals refer to strains. For example, “1” is the strain file for strain 1.

1 \

==merge==> 1,2 \ / ==filters==> SNPs

2 / \ /

==merge==> 1,2,3,4 ==find SNPs==> SNPs (text)

3 \ / \

==merge==> 3,4 / \ ==findGenes==> Genes

4 /

In the second case the user selects two sets of strains, and a read frequency threshold for each. Each set is merged into a consensus. These are compared to find SNPs between them, as well as the major allele, major product, minor allele, minor product and other statistics.

set 1: 1 \

==merge==> 1,2 ==find consensus==> consensus \

set 1: 2 / \

==find SNPs and major alleles==> SNPs

set 2: 3 \ /

==merge==> 3,4 ==find consensus==> consensus /

set 2: 4 /

## == Sample search runner script ==

The following is a sample script generated to run a specific search, with specific parameter values provided by the user. It searches for Genes that contain SNPs found in a specified group of strains. The user has selected:

* 9 strains (with internal IDs 13,21,16,5,29,3,6,11 and 8)
* a read frequency threshold of 20 and some other parameters.
* a 0% minimum polymorphism percent (ie, if there is any polymorphism at a location, the SNP is kept)
* an allowance of up to 8 unknowns for a SNP to be kept.
* additional parameters to control the filtering of Genes based on a Gene's SNP characteristics (eg, SNP density)

set -e

set -x

cd $HSSS\_WORKING\_DIRS/hsssFindSnpsInOneSet.1408730457121

mkfifo fifo1 fifo2 fifo3 fifo4 fifo5 fifo6 fifo7 fifo8 fifo9

trap "rm fifo1 fifo2 fifo3 fifo4 fifo5 fifo6 fifo7 fifo8 fifo9 " EXIT TERM

hsssMergeStrainsWrapper fifo1 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/13 13 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/21 21 &

hsssMergeStrainsWrapper fifo2 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/16 16 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/5 5 &

hsssMergeStrainsWrapper fifo3 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/29 29 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/3 3 &

hsssMergeStrainsWrapper fifo4 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/6 6 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/11 11 &

hsssMergeStrainsWrapper fifo5 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/8 8 fifo1 0 &

hsssMergeStrainsWrapper fifo6 fifo2 0 fifo3 0 &

hsssMergeStrainsWrapper fifo7 fifo4 0 fifo5 0 &

hsssMergeStrainsWrapper fifo8 fifo6 0 fifo7 0 &

hsssFindSnpsInOneSet fifo8 $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/referenceGenome.dat 9 0 8 | hsssGeneCharacteristicsFilter $HSSS\_FILES/EhistolyticaHM1IMSS/readFreq20/contigIdToSourceId.dat geneLocations.txt all 8 -1 0 -1 0 100011 > $HSSS\_WORKING\_DIRS/hsssFindSnpsInOneSet.1408730457121/results

exit