The structure of Trait-o-matic

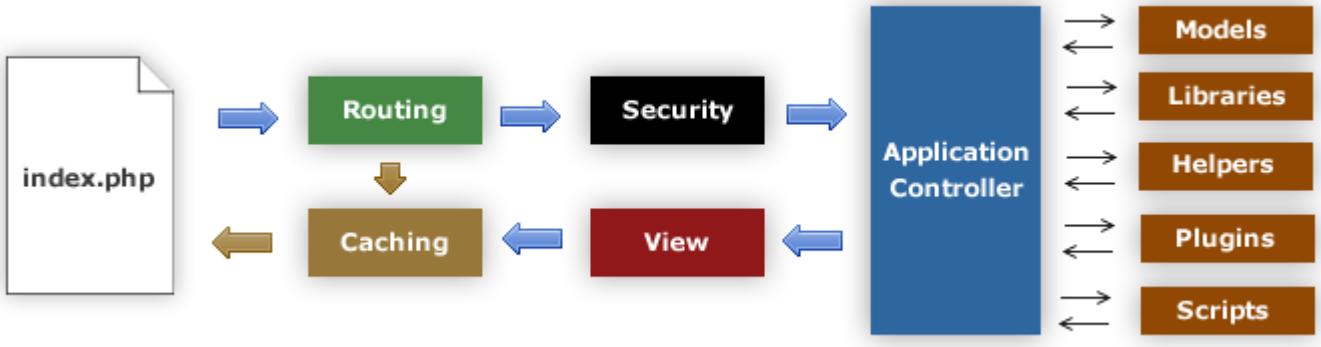
Scripts:

Python
Trait-o-matic uses python scripts for multiple purposes. One of the most important purposes is to parse the SNPedia and HapMap data into a computer-friendly format.

PHP

-CodeIgniter Frontend

Trait-o-matic uses CodeIgniter as a front end to help manage the many resources of the web service. CodeIgniter is a Web Application framework written in PHP that allows us to abstract away from coding individual pages on the website, and allows for a focus on creating libraries and models that run in the background. It follows the general model of **Model-View Controller-View.** Thus, we can spend little time worrying about the presentation of the web service and more on the actual functionality of the program modules.



http://codeigniter.com/user_guide/overview/appflow.html

Services

mySQL
mySQL is a very popular relational database system. Relational means that it contains relations between entries and particular attributes (think of this as one particular row in an excel spreadsheet). Trait-o-matic contains three such databases, with their contents described below.

Apache
This is the web server that the traitomatic service resides on. It allows anyone to utilize the trait-o-matic resources and manages users.

Database Contents

DATABASE	TABLE	ATTRIBUTE	DESCRIPTION	
ariel				
	evidence			
		phenotype		***Note: this table is currently empty!***
		chr		
		start		
		end		
		allele		
		inheritance		
		references		
		tags		
		id		
caliban				
	hapmap27		This is the HapMap Allele data	
		rs_id	The ID of this snp *rs12241440*	
		chr	chromosome	
		start	start location	*note: the start and end are always 1 ap*
		end	end location	
		strand	strand (+ or -)	
		pop	population descriptor. Values: ASW: African ancestry in Southwest USA, CEU: Utah residents with Northern and Western European ancestry from the CEPH collection, CHB: Han Chinese in Beijing, China, CHD: Chinese in Metropolitan Denver, Colorado, GIH: Gujarati Indians in Houston, Texas, JPT: Japanese in Tokyo, Japan, LWK: Luhya in Webuye, Kenya, MEX: Mexican ancestry in Los Angeles, California, MKK: Maasai in Kinyawa, Kenya, TSI: Toscans in Italy, YRI: Yoruban in Ibadan, Nigeria.	
		ref_allele	The reference allele at this point (A,T,G,C)	
		ref_allele_freq	The frequency of the reference allele	
		ref_allele_count	The number of examples of reference allele	
		oth_allele	The other allele at this location (A,T,G,C)	
		oth_allele_freq	The frequency of the other allele	
		oth_allele_count	The number of examples of the other allele	
		total_count	The total number of genotypings of this locus	*Note: this could be used to get some so*
	morbidmap		This is the OMIM Morbid Map database, which contains the cytogenetic map location of disease genes described in OMIM	
		disorder	The name of the associated disorder	
		symbols	The gene symbols associated with a disorder	
		omim	The oMIM ID of the variant, *ie 100650*	*Note that this is just the base number*

	location	The chromosomal location of said variant, *ie 10q25-q26*	
omim		This is the OMIM database, which contains particular alleles that are associated with disease	
	phenotype	The name of the associated disease	
	gene	The short name of the gene, ie HPRT	
	amino_acid	The amino acid substitution *ie Glu-Met*	
	codon	The number of the codon that contains the substitution	
	word_count		*Note: Don't know what this is!*
	allelic_variant_id	The OMIM ID of the variant, *ie omim:100650.0001*	
refflat		This is the UCSC Genome browser's reference genome. It contains the names and locations of 32,494 human genes	
	geneName	The short name of the gene, *ie HPRT*	
	name	The NCBI reference name *ie NM_032172*	
	chrom	The chromosome of the gene *ie chr7*	
	strand	The positive (+) or negative (-) strand	
	txStart	The start locaiton of the transcript	
	txEnd	The end location of the transcript	
	cdsStart	The start location of the coding sequence	
	cdsEnd	The end location of the coding sequence	
	exonCount	The number of exons	
	exonStarts	The start locations of the exons	
	exonEnds	The end locations of the exons	
snp129			***Note: This is currently empty!***
	bin		
	chrom		
	chromStart		
	chromEnd		
	name		
	score		
	strand		
	refNCBI		
	refUCSC		
	observed		
	molType		
	class	What class of SNP this is. *Possible values:*	'unknown','single','in-del','het','microsa
	valid	What information was used to confirm this SNP. *Possible values:*	unknown','by-cluster','by-frequency','b
	avHet		
	avHetSE		
	func		
	locType		

		weight		
	snpedia			***Note: This table is currently empty!***
		phenotype		
		chr		
		start		
		end		
		strand		
		genotype		
		pubmed_id	The pubmed ID of this SNP.	
		rs_id	The ID of this snp *ie rs12241440*	*Note: Reference with the data from Hap*
dbsnp				
	OmimVarLocusIdSNP		This is the OMIM database in another format.	
		omim_id	The ID of a variable locus in OMIM *ie 100690*	
		locus_id		
		omimvar_id	The ID of a particular variation in OMIM *ie 0001*	
		locus_symbol		*Note: this attribute is always NULL in ou*
		var1		*Note: this attribute is always NULL in ou*
		aa_position		
		var2		
		var_class	the class of variation	*Note: this attribute is an INT and there numbers*
		snp_id	the ID of the SNP	*Note: Check whether this ID is related to*
	b129_SNPChrPosOnRef_36_3			
		snp_id	Internal SNP Id	*Check whether this relates to rs_id or sn*
		chr	Chromosome -- *ie 6*	
		pos	Position along this chromosome	
		orien	Orientation	*Check whether this orientation is correc*
		neighbor_snp_list	?	*Note: this attribute is always 0 in our da*
		is_par	Either NULL or Y.	*Check what this means*

Accessing trait-o-matic from a shell:

If you are interested in exploring trait-o-matic yourself, then follow these instructions:

1. Request access for a control panel to the VM at https://controlpanel.freelogy.org/users/signup
2. SSH into your shell (for Harvard student, you can use ice3.fas.harvard.edu with your FAS username and password, the same password you use for email)
    a. To do this on a mac, open up Terminal, and then type *ssh ice3.fas.harvard.edu*
    b. To do this on a mac, download putty.exe (google for it) and then connect to ice3.fas.harvard.edu using the interface
3. Generate an RSA keypair by executing the command *ssh-keygen –t rsa*
    a. When prompted where to save the file, just hit enter (this will use the default)
    b. When prompted to enter a password, enter a password you can remember twice. This is what you will use to connect to the VM.
4. Open your ~/.ssh/id_rsa.pub file (for example, by typing *nano ~/.ssh/id_rsa.pub*)
5. At the control panel above, click on ssh keys on the top, and then Add Key. Then, paste the entire contents of the id_rsa.pub file into this file.
6. On the main page of the control panel, you will see a few modifications that you will have to make to your

Troubleshooting:

Email info@scalablecomputingexperts.com