Brief description

Aim

The aim of the virome dark matter project was to (a) catalogue the diversity of viruses present in humans and (b) learn features from the large proportion of viral reads that cannot be annotated (i.e. dark matter) using standard approaches. Finally, we would use the features from the dark matter virome to discover novel unknown viruses and with a larger repertoire of viruses, more comprehensively catalogue and understand the human virome.

Approach used (More in the sections to follow, along with where code and dataset used is kept on microb120 server)

We collected >900 shotgun sequenced datasets that were purified for viruses, and used a very early version of the Sunbeam pipeline to process and assemble reads and call contigs. Then we use getorf tool to find ORFs on all contigs, and use (a) blastn on nt database to annotate all ORFs and contigs, (b) blastx on viral protein specific Skip Virgin's database to annotate all ORFs, (c) hmmscan from HMMER on pFAM and vFAM databases to annotate all ORFs.
Then I find all ORFs that have no annotation, and use cd-hit tool to remove redundant ORFs (aka remove which have 80% or more similarity). Next, I blast all ORFs against each other and use the mcl tool to cluster them, align sequences in each cluster (via clustal omega) and make hmms from the multiple alignment (via hmmbuild from HMMER). I then iteratively merge these dark matter HMMs into a more compact database.
I then use these dark matter HMMs to annotate all ORFs again, and use the 3 following strategies to further understand and illuminate the dark matter HMMs
(a) Find ORFs that can be annotated with both known viral protein and dark matter viral HMM. This suggests that the dark matter viral HMM is the distant homolog of the known viral annotation, and can now be used to annotate and understand many more viral proteins (that were previously unannotated and hence dark)
(b) Find contigs with some ORFs having a known viral annotation and some ORFs having a dark matter HMM annotation. This suggests that the dark matter HMM family belongs to a specific known virus, and can be used to further annotate and understand the previously dark viral protein matter
(c) Find multiple dark matter HMMs that often annotate ORFs, which co-occur on the same contig. This suggests that these dark matter HMMs belong to a novel virus.

Current Status

The virome reads have been processed and contigs/genes have been called on it. The contigs and the ORFs have been further annotated using Blast nt and a viral protein specific dataset. The ORFs have been annotated with pFam and vFam databases.

The unannotated ORFs have been collected, and clustered together (using cdhit, blast all and mcl tools). Each cluster with unannotated ORFs (aka the dark matter ORFs) has been used to learn a specific dark matter viral HMM. I have then iteratively merged these initial dark matter viral HMMs into a more compact dataset of dark matter HMMs (dvmFAM).

I have use these dvmFAMs to annotate all ORFs again (and compare to existing annotations if any) to better understand the features of the viral dark matter. I find ORFs that could be annotated strongly with a known viral annotation and weakly with the dvmFAM, suggesting that the dark matter HMM family is a distant homolog of that known viral protein. This in turn would allow for annotation of ORFs that were previously dark matter, and could only be annotated with the dvmFAM.
I also find dark matter viral proteins (via dvmFAM) that often co-occur on the same contig as the known viral proteins, making it possible to annotate these dvmFAMs as belonging to specific viruses.
Finally, I find examples of dark matter proteins (via dvmFAM) often co-occurring together on a contig, suggesting the presence of novel viruses and potential cassettes.
These last 3 steps of understanding features from dark matter viral HMMs, is a work in progress and needs additional clearly defined experiments to make a story out of it, and not just a hodgepodge of examples.

Salient features of a potential manuscript and current status

- Major Points
    - Algorithm for viral dark matter discovery, and dvmFAMs
        - Status: Complete
    - Features of viral dark matter
        - Status: Need to follow up more on annotation points described above. I have specific examples but need to systematically sift through this,
    - Landscape of viral dark matter across samples and different groups
        - Status: Need to plot figures to show some dark matter ORFs present across multiple samples, while some new. No new data needs to be generated. For comparison with other groups, need to find if our dark matter HMMs (dvmFAM) present in them too (or dark matter is unique to our lab). Here we need to run our dark matter HMMs on any virome dataset (make from Skip Virgin's group).
- Minor Points
    - Description of known and dark matter virome across tissues and disease samples and controls
        - Status: Need to draw PCA and t-SNE etc plots. Underlying data exists and no new data needs to be generated for this

Potential Future experiments

1. Run dark matter HMMs on reads (as opposed to just ORFs on few contigs) to better catalogue and understand the diversity of human virome.
2. Run dark matter HMMs on whole genome sequenced viruses (which we can get experimental access to in future) and find if some dark matter HMM annotates an ORF there. Now, we can perturb and study that novel dark matter family.

3. Find if the dark matter viral proteins leave a signature on our immune system (IGG and IGA sequencing etc type experiments)

Details of dataset, software versions, databases and code

Dataset

I used 923 shotgun sequenced viral purified samples for my meta analysis. They come from different tissues (skin, lung, gut) and different hosts (human, chimp, fly), without positive (spiked in) and negative controls. The samples have been sequenced both on miseq and hiseq platforms.

The details of all the samples is in the "Samples registry master file.xlsx". It has information on Sample Type (ex. BAL fluid, Pre-wash, Stool, Buffer), Disease status (if any), Tissue of interest (BAL implies Lung, Stool implies Gut, or positive/negative control), Host, Sample Id (unique identifier), Subject id (for time point samples where the subject remains the same), Control description (if any), Additional comments (if any), Data generated by lab member, Sequencing platform, Date of sample collection, Library Prep method, cDNA or DNA, Part of which study (Sarcoid, SCID, FMT, Guanxiang Fly etc)

All the raw compressed fastq files (the two paired reads) are kept in microb120 at /media/THING2/avarun/virome_datasets

Databases

The databases used in the analysis are kept in microb120 at /media/THING2/avarun/databases.

The blast databases are inside the folder genomeIndexes/blast

The nt blast files has been downloaded using the download.files code (December 22nd version, current one is April 2018), so this is definitely dated.

The viral specific database is from Skip Virgin's group and it is Feb 25th 2014 version floating in the lab. It is kept inside the folder 02252014_Virgin_VLP_DB

The pFam database (version 31.0) is kept in the folder Pfam31.0. The main database file is Pfam-A.hmm, but we have added some custom HMMs to it too. Details are in the README file in the same folder.

The vFAM database is kept in the folder vfam. We use the vFam-B_2014.hmm (more comprehensive) database for all our analysis. Details of sequences that went inside each vFAM is kept in the annotationFiles_2014 folder.

The dark matter HMMs are kept in the folder dmvFAM. All the original dark matter HMMs are in files beginning with dmvFAM_all_nonannot.hmm. I then merge some of the dark matter HMMs and the final version is in files dmvFAM_all_nonannot_new_collapsed.hmm.

Software Versions

For Blast I used the version 2.2.0

For HMMER suite I used the version 3.1b2

For MCL (clustering) I used the version 14-137
For getorf (gene calling) I used the EMBOSS version 6.6.0 suite
For cdhit I used the version 4.6.1
For sunbeam, I used the version 0.1.1 (only the read processing and contig building steps)

Code and Pipeline

All the code is kept on microb120 in the /media/THING2/avarun/sunbeam folder
Read processing and Contig Building

- We first use the sunbeam pipeline to do read processing and contig building by using collection specific yml files. These are kept in the folder /media/THING2/avarun/sunbeam/sunbeam/samples_config_files_output/ and has multiple yml files for different collections (chimp_fecal, christel_fmt, ctot_acr etc)
    - The output from this is saved in folders /media/THING2/avarun/sunbeam/sunbeam_chimp_fecal/, sunbeam_ctot_acr_bal/ etc.
        - All the decontaminated and processed reads (R1 and R2 fastq) are kept inside these folders in the directory qc/decontam-phix.
        - The assembly information for each sample is kept in the sample specific directory assembly/SAMPLEID_assembly. For all my analysis, I use contig.fa file

ORF calling and Annotation methods

- Now that we have the contigs, I call ORFs on them using the getorf utility. I only call ORFs of size 150 AA or more and which are between start and stop codons (in all 6 frames, 3 forward and reverse)
- I also call Blastn on nt and Blastx on the Skip Virgin's database, both with cutoff of E value of 0.001
- I also run hmmscan with pFam and vFam databases on all ORFs.
- All the code for this is kept in the folder /media/THING2/avarun/sunbeam/CODE/find_profile_hmm_blast_orfs. The files used are find_orfs_blastx_pfam_between_start_stop.py, find_blastn_between_start_stop.py
- The annotations for each sample is kept in the directory /media/THING2/avarun/sunbeam/COLLECTION_ID/annotation/{blastn/nt/contig/ or in genes} directory.

Alignments and Coverage

- This is not central to any results, but the code for aligning reads to contigs and orfs is kept in the directory /media/THING2/avarun/sunbeam/CODE/find_alignments_coverage.
- The code is in the file align_reads_coverage.py, and we use bowtie, samtools and bedtools to find reads mapping to each contig, their counts and coverage
- The resulting counts and coverage files for each sample is kept in the qc directory /media/THING2/avarun/sunbeam/COLLECTION_ID/qc/decontam-phix

Positive Controls as set of all viruses

- I downloaded all the NCBI viruses (more detail in the README file /media/THING2/avarun/sunbeam/reference_viral_genomes/README), along with their taxa and id information.
- I then find ORFs and annotations (like above for all viruses). Details are in the files find_reference_virus_blast.py, find_reference_virus_orf.py, find_reference_virus_profilehmm.py and find_reference_virus_profilehmm_wrapper.py in the directory /media/THING2/avarun/sunbeam/CODE/find_profile_hmm_blast_orfs
- The resulting ORFs and annotations are saved in the /media/THING2/avarun/sunbeam/reference_viral_genomes/annotations directory

Compile annotations

- Now that we have annotations for each contig and ORF, and also the reads mapping to it, we create summary files for each sample, where annotation information on each ORF and contig is saved. We only work with ORFs of length 300 or more (even though we called ORFs of length 150 aa or more)
    - We use the files master_table_orf_annotations.py, master_table_orf_annotations_stringent.py, master_table_orf_stats.py, master_table_orf_stats_stringent.py in the directory CODE/compile_annotations to process all the raw annotations.
        - For stringent annotations we use the cutoff of E value of 1e-10, and for pFam and vFam to be the E value of 1e-10 and bit score of 30
        - The annotations for each sample are saved in the respective directory /media/THING2/avarun/sunbeam/COLLECTION_ID/SAMPLE_ID_{orf_stats/orf_stats_stringent/all_orf_contig_annotation/all_orf_contig_annotation_stringent}
        - We use similar files to compile annotations on all reference viruses master_table_orf_annotations_stringent_reference_virus.py, master_table_stats_stringent_reference_virus.py and save the resulting compiled stats in the CODE/compile_annotations/data_merit directory

Dark Matter Finding

- We find all unannotated ORFs using the file find_annotated_unannotated_orfs.py in the directory /media/THING2/avarun/sunbeam/CODE/dark_matter_new/.
- Using file find_orfs_unannotated_seq.py we then find the fasta sequences of the unannotated ORFs and save them here data_merit/all_orf_unannotated_aaseq.fa
- I then use cdhit to cluster all unannotated ORFs with 80% or more similarity and 90% or more alignment. Details in the file cdhit_blast_mcl.sh
- I then make a blast database of representative sequences from cdhit and then divide the cdhit file into smaller files, to blast all against this database.
    - Details of blast database are in the cdhit_blast_mcl.sh file
    - Details of subdivision of cdhit ORFs into smaller files and then blast against the database created is in divide_cdhit_smallfiles_blast.py file. All the resulting files and blast results are saved in the directory data_merit/blastall_results
- Now I cluster these blast all results using the mcl program (via mcxload, mcl and mcxdump utilties)

    - Details of script in the file cdhit_blast_mcl.sh
- After getting the clusters, I only work with those that have 3 or more ORFs in it and make multiple alignments of it, and finally HMM from it. Details are in the subdivide_makehmm_clusters.py file and resulting clusters and their individual HMMs are kept in the data_merit/mcl_clustering/clusters directory (for each cluster directory)
    - We next combine these initial dark matter HMMs into one big file using the script combine_hmms_onedb.py. I then delete this file and instead save its HMMER compliant database file in the directory /media/THING2/avarun/databases/dmvFAM and in the file dmvFAM_all_nonannot.hmm.XYZ
- The clusters used can still be coarse and I further try to narrow them down. So I try to find dark matter HMMs which consistently annotates the same unannotated ORF from cdhit. I then tabulate such dark matter HMM pairs and collapse some of these clusters (aka HMM into a new cluster).
    - I use the script find_darkmatter_annot_cdhitall.py to find the annotations of dark matter HMMs on all ORFs. I them use the script annotations_cdhitall_darkmatter.py to tabulate which HMMs often annotate the same ORFs
- Following this I use mcl utilities again to merge such clusters and then like before align the ORFs in each cluster and make HMMs out of it. I then combine these HMMs and save the HMMER compliant databases only.
    - The code for mcl utilities is kept in cdhit_blast_mcl.sh file. The code for merging the old clusters, multiple alignment and then making HMMs is in subdivide_makehmm_collapsedclusters.py file. I use the combine_hmms_onedb.py to merge individual HMMs together, and save the resulting database in the directory /media/THING2/avarun/databases/dmvFAM and in the file dmvFAM_all_nonannot_new_collapsed.hmm.XYZ. The new collapsed clusters with each individual HMM are kept in the directory data_merit/mcl_clustering/new_collapsed_clusters

This is the limit of the useful code that someone can use to find all the dark matter HMMs (the latest version (dmvFAM_all_nonannot_new_collapsed.hmm).
I have a lot of analysis scripts, which are not in a stable version because the analysis is not at a steady finalized state.
However, post this point, I annotate all ORFs with dark matter HMMs, and try to do the 3 illumination steps discussed in the very first section of approach.