Searching the Human Genome with a simple Distributed System

Stephen Tredger
University of Victoria
Victoria, British Columbia

I. Introduction

Since the 1970s when viral DNA, or fragments that resemble viral DNA were found to exist in the human genome, researches have been investigating how it got there, and what role if any it plays in development. Such viral elements are known as endogenous viral elements and the most well known in humans come from a family of viruses called retroviruses. What makes retroviruses unique is that they essentially incorporate their own DNA into that of its host through a process called reverse transcription. The most well known human infecting retrovirus is the human immunodeficiency virus (HIV) which is classified as an exogenous retrovirus. Retroviral elements that are vertically transmitted (passed to siblings) in human DNA are known as human endogenous retroviral elements (HERVs) and it is estimated that 4-8% of the human genome is composed of HERVs [1]. While these HERVs are the most common viral elements found in our genome, other viral elements have been shown to exist in animal genomes and are generally called endogenous viral elements (EVEs). Most EVEs are quite modified from the origin fragment, but this leads to an interesting question on how easy it is to find EVEs in our genome? Can we find any suspect sequence with a simple distributed system?

II. Experiment Framework

In this case genome searching embarrassingly parallel. We want to compare viral genomes to ours, so naturally each virus comparison doesnt care about the state of the other comparisons are doing. In this sense we can break up our job into many sub jobs and run them independently on several machines. The experiment environment uses the Geni Experiment Engine to provide three nodes for computation [2]. The nodes are controlled through a master using the python Fabric module. Fabric essentially allows shell commands to be run on remote machines through a python interface, and makes managing remote machines very simple. I used the SageFS distributed filesystem to manage files across the nodes. The SageFS backend exists on three clusters of machines on the Savi research network. One cluster is in Victoria, one in Carlton, and the third in Toronto. I used two tools to actually perform sequence alignment, Bowtie2 and MUMmer. Bowtie2 is built to align sequences to a reference genome very quickly using a genome index [3]. Bowtie2 expects the reference genome to be in a binary index format that it can understand, so for these experiments I used the Bowtie2 index of the human genome built from the hg19 build of the human reference genome available on the Bowtie2 sourceforge site. MUMmer is built to align large sequences to each other and simply takes a reference genome and sequences to align as basic arguments [4]. The reference human genome was the GRCh38 build the NCBI database [5]. Where as Bowtie2 could align to the entire set of human chromosomes at the same time, MUMmer was run differently in that each viral sequence was run against each human chromosome separately. The viral genomes also came from the NCBI database.

III. Experiment Setup

I wrote a simple web crawler in python that crawled the NCBI web interface and downloaded all virus genomes then stored them in the filesystem. Since the filesystem is already physically partitioned into three locations, I decided to have three crawlers running in parallel, one on each of the GEE nodes, and distribute the work so that approximately one third of all the genomes would end up at each location. The end result was that Toronto ended up housing 1864 genomes, Carlton 1729, and Victoria 1942. The split was not entirely symmetrical as the number of links to follow was partitioned, not the genomes themselves, and any given link could contain a genome, multiple genomes, or not contain any at all. The crawler was also used to download the GRCh38 reference sequence from the NCBI database. Since the data was already partitioned into separate locations, each machine in the experiments that follow worked exclusively on virus genome that was present at one given location, and uploaded results back into the filesystem at the same location.

IV. Bowtie2

A. Experiment

I wrote a simple python script that would grab a list of all the virus genomes at a given location (either Victoria, Toronto, or Carlon in this case), then download the genome and align it against the reference index using Bowtie2. Each node had to have a local copy of the reference index and so I got the script to grab it from the Bowtie2 sourceforge site. I could have stored the reference in the distributed filesystem, but the reference index is 3.5 GB and there was not enough space in the filesystem to store it. Normally 3.5 GB would not be a problem but the filesystem is a prototype and I wanted it to have a minimal footprint on the machines where it runs.

After the reference had been downloaded the node could start processing. For a given virus genome the node would grab it out of the sage filesystem and transfer it to the local filesystem (a function provided with sage), as the sage client is simply a python interface to the remote location and Bowtie2 is a standalone binary. Then Bowtie2 was run with default local scoring parameters looking for a local alignment comparing the virus sequence to the human sequence. As a sight aside, the memory on the nodes had to be increased from 512 MB to at least 4 GB to run Bowtie2. The memory was actually increased to 8 GB on all the nodes as MUMmer required more than 4 GB. After the alignment was finished the output file was placed back into sage. It took upwards of 36 hours to to align all 5534 sequences and produce the results. Bowtie2 reports if a suitable match was found, and has more information on the match about where it occurs and what the alignment looks like. The results found by Bowtie2 are summarized in Table I and presented in Figure 1.

B. Results

Of the 5534 alignments performed only 36 probable alignments were found. Of the 36 none were particularly long alignments, the largest being HERV-K113 with approximately a quarter of the viral sequence aligned to chromosome 4. This is somewhat surprising as HERV-K113 is only found within the human genome. Most of the alignments are likely false positives not actually representing gene transfer as for such small sequences It is far more likely to be random chance. This may seem like somewhat of a negative result, but it really shows that EVEs are more elusive than I previously thought. This is not amazingly surprising as many of these fragments have existed in our genome for millions of years, so it is not hard to imagine the sequences have diverged over time. Additionally the places where these fragments are inserted are somewhat volatile. In fact if such elements are inserted into critical genes, a disease state would likely occur. From the results obtained and the knowledge that the sequences are most likely not conserved I decided to look at a more sensitive alignment using MUMmer.

V. MUMMER

I modified the python script to use MUMmer instead of Bowtie2 as MUMmer showed some matching sequence even if no matches were present in the overall alignment. MUMmer reports sequences larger than a given minimum size, in this case 20 bp, so even in highly conserved sequence some minimal matches are likely to be reported. The job now required the reference sequence to not be in an index so the crawler was used to grab the GRCh38 reference sequence from the NCBI database into the filesystem. To begin the alignments each node grabs a full copy of the chromosome reference set and stores it locally. Then like in the Bowtie2 experiment each node is responsible for a partition of the viral genomes. However unlike the Bowtie2 experiment each genome must be aligned to each chromosome separately. The reference set of chromosomes contains chromosomes 1 - 22, an X and Y, as well as the mitochondrial dna (MT), and some unmapped sequence (UT). This means that each genome is aligned against 26 chromosome sequences. Once an alignment has completed the output file is uploaded back into the sage filesystem. The job was started and from observation it takes approximately 2 minutes to perform an alignment. Unfortunately there are 5534 viral sequences and 26 chromosomes, so about 143884 alignments must be made. Dividing the work over three machines at 2 mins per alignment means the job will take approximately 95922 minutes, about 67 days! In order to get the alignments within a day I would need around 200 machines each with over 4 GB of ram. I decided to not complete the job as the time commitment on the GEE machines was too large.

VI. CONCLUSIONS

The simple distributed system I set up to search genomes required only a basic knowledge of python, and unix systems. The system consisted of three GEE nodes to perform computation, my local machine to act as the controller, and an instance of the sage filesystem to store files. The actual system was built using the fabric python library and the sage filesystem. I used a web crawler written in python to pull genomes into the sage filesystem, which were then used by the actual sequence alignment tools Bowtie2 and MUMmer. Distribution of the jobs was done using fabric to run jobs on the remote GEE nodes.

I aligned 5534 viral DNA sequences to the hg19 build of the human genome using Bowtie2, ending up with 36 reported matches. Of all the matches only the HERV-K113 genome is likely to be an actual match as about one quarter of the total sequence was aligned. The lack of concrete alignments is not particularly alarming as most of the viruses aligned do not infect humans, are not retroviruses, and most human discovered EVEs have existed in the human genome for millions of years, an ample amount of time for current viruses to significantly diverge.

REFERENCES

[1] A. Katzourakis and R. J. Gifford, “Endogenous viral elements in animal genomes,” *PLoS Genet*, vol. 6, no. 11, p. e1001191, 11 2010.

[2] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, “Geni: A federated testbed for innovative network experiments,” *Computer Networks*, vol. 61, no. 0, pp. 5 – 23, 2014, special issue on Future Internet Testbeds Part I.

[3] B. Langmead and S. L. Salzberg, “Fast gapped-read alignment with bowtie 2,” *Nature methods*, vol. 9, no. 4, pp. 357–359, 2012.

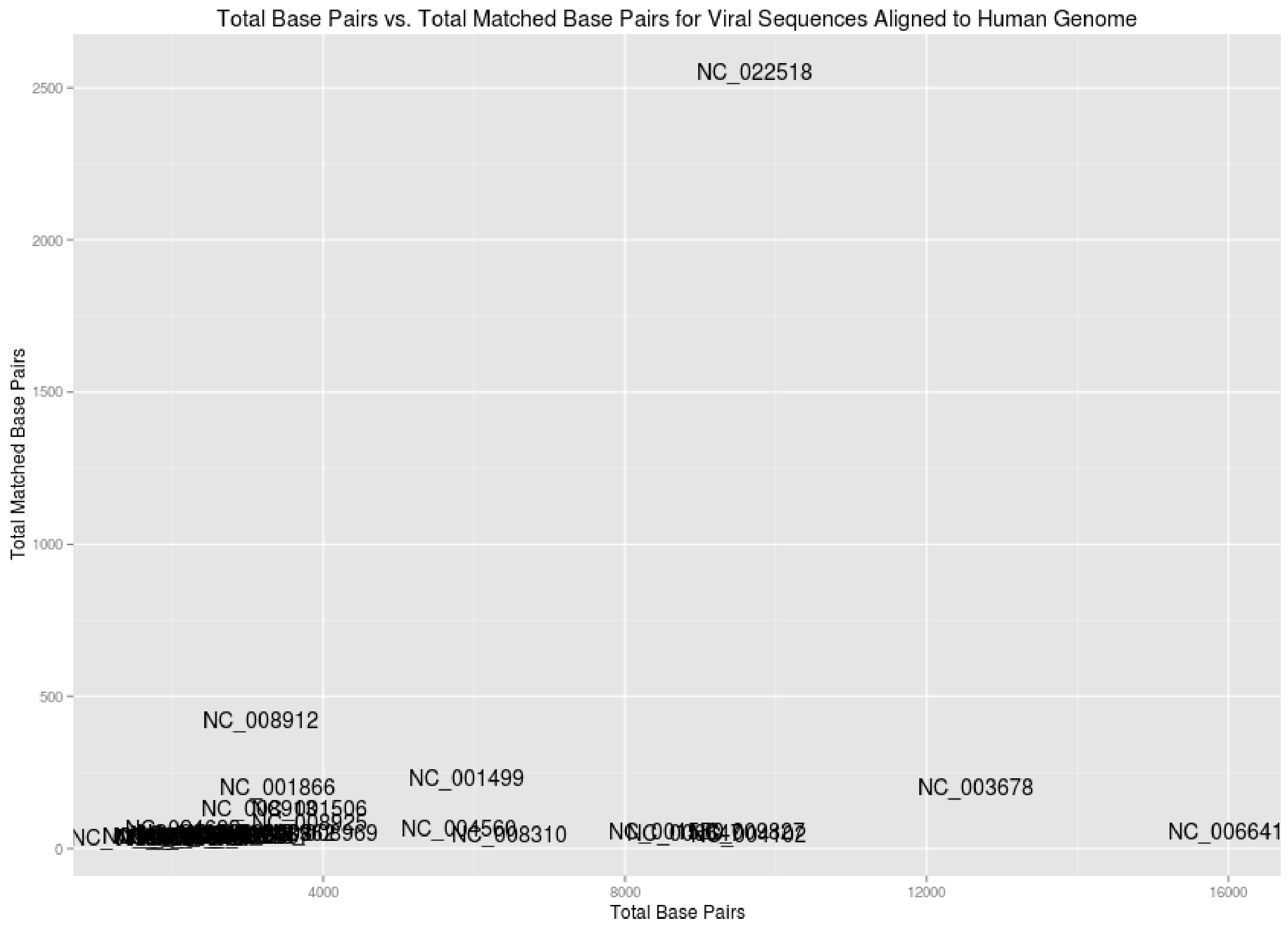
[4] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg, “Alignment of whole genomes,” *Nucleic Acids Research*, vol. 27, no. 11, pp. 2369–2376, 1999.

[5] L. Y. Geer, A. Marchler-Bauer, R. C. Geer, L. Han, J. He, S. He, C. Liu, W. Shi, and S. H. Bryant, “The ncbi biosystems database,” *Nucleic acids research*, p. gkp858, 2009.

TABLE I
MATCHING SEQUENCE FOUND USING BOWTIE2.

NCBI Name	Ref.	Chrom.	Max Match	Tot Match	Tot Seq
Abelson_murine_leukemia_virus_uid14654	NC_001499	chr9	207	232	5894
Alternaria_alternata_dsRNA_mycovirus_uid30367	NC_010989	chr21	41	41	2794
Alternaria_alternata_dsRNA_mycovirus_uid30367	NC_010991	chr5	37	37	1420
Aspergillus_foetidus_dsRNA_mycovirus_uid186431	NC_020101	chr1	41	41	2466
Aspergillus_foetidus_dsRNA_mycovirus_uid186431	NC_020102	chr5	41	41	2005
Avian_myelocytomatosis_virus_uid14909	NC_001866	chr8	151	203	3392
Broad_bean_mottle_virus_uid14833	NC_004006	chr3	40	40	2293
Cassia_yellow_blotch_virus_uid15419	NC_007001	chrX	42	42	2091
Cotesia_congregata_bracovirus_uid14556	NC_006641	chr2	59	60	15960
Cotesia_congregata_bracovirus_uid14556	NC_006647	chr3	52	52	8785
Cowpea_chlorotic_mottle_virus_uid14758	NC_003542	chr5	39	39	2173
Dill_cryptic_virus_1_uid225921	NC_022614	chr9	39	39	2013
Dill_cryptic_virus_1_uid225921	NC_022615	chrX	33	44	1837
Dill_cryptic_virus_2_uid198774	NC_021148	chr6	44	44	2354
Glypta_fumiferanae_ichnovirus_uid18767	NC_008862	chrX	51	52	2299
Glypta_fumiferanae_ichnovirus_uid18767	NC_008895	chr5	45	45	2854
Glypta_fumiferanae_ichnovirus_uid18767	NC_008908	chr22	53	55	3084
Glypta_fumiferanae_ichnovirus_uid18767	NC_008912	chr2	181	423	3179
Glypta_fumiferanae_ichnovirus_uid18767	NC_008913	chr3	86	134	3157
Glypta_fumiferanae_ichnovirus_uid18767	NC_008925	chr12	82	90	3821
Groundnut_ringspot_and_Tomato_chlorotic_spot_virus_reassortant_uid66459	NC_015467	chr12	41	41	3067
HCBI8_215_virus_uid257701	NC_024689	chr5	69	69	2152
Hepatitis_C_virus_genotype_6_uid20939	NC_009827	chr17	58	58	9628
Hepatitis_C_virus_uid15432	NC_004102	chr14	47	47	9646
Hibiscus_latent_Singapore_virus_uid17573	NC_008310	chr5	49	49	6474
Human_endogenous_retrovirus_K113_uid222261	NC_022518	chr4	1222	2552	9716
Hyposoter_fugitivus_ichnovirus_uid18779	NC_008962	chr12	55	55	3385
Hyposoter_fugitivus_ichnovirus_uid18779	NC_008969	chr14	46	51	3958
Jingmen_Tick_Virus_uid247973	NC_024112	chr6	43	53	2850
Mason_Pfizer_monkey_virus_uid14683	NC_001550	chr15	59	59	8557
Murine_osteosarcoma_virus_uid14655	NC_001506	chr11	134	135	3811
Oyster_mushroom_spherical_virus_uid14951	NC_004560	chr7	41	69	5799
Pestivirus_Giraffe_1_uid14780	NC_003678	chr12	163	202	12646
Phytophthora_infestans_RNA_virus_1_uid40329	NC_013221	chr4	41	41	2896
Pleurotus_ostreatus_virus_uid15169	NC_006960	chr3	47	47	2223
Raspberry_latent_virus_uid56055	NC_014602	chr10	43	46	2565

Fig. 1. Plot of Matching Sequence vs Total Sequence


Total Base Pairs vs. Total Matched Base Pairs for Viral Sequences Aligned to Human Genome
Total Matched Base Pairs
Total Base Pairs
NC_022518
NC_008912
NC_001866
NC_001499
NC_003678
NC_004560
NC_008310
NC_006641