The Viral Cloud Resource

Ravi Sanka, Ntino Krampis, Alex Richter, Andrey Tovchigrechko

@ GCC 2014, Baltimore

J. Craig Venter™
INSTITUTE

VCR Pipeline

- Objective: To produce an interactive and informative display of an annotated viral genome based on provided sequences, both reads and established references.
- Inputs
  - Read sequences that can come from a variety of platforms.
  - A viral database chosen from a provided list of organisms.

J. Craig Venter™ INSTITUTE

VCR Pipeline

- Outputs:
  - Assembled genome of the viral database's organism.
  - Standard output of the JCVI VIGOR tool.
  - An interactive JBROWSE visualization of the annotated genome.
- Consists of two stages (Galaxy tools)
  - Viral Assembly
  - JCVI VIGOR

Viral Assembly

- What is it?
  - A python script that takes in provided read sequences, established reference database file of the target viral genome, and produces a FASTA file of the viral genome modified by the read sequences.
  - The means of achieving this goal were determined by the JCVI Viral Group.
- The program achieves this through the following steps.

Viral Assembly

- Step 1 – Acquire Inputs
  - Input read sequences are retrieved via option parsing. User can provide and combination of 454 (sff), Illumina (fastq), Sanger (fasta and qual), and IonTorrent (fastq).
  - The name of the chosen reference is also retrieved via option parsing. The name points the script to the appropriate, preset location of the BLAST-able database.
  - Read sequences are converted to FASTQ, offset-33 format.

Viral Assembly

- Step 2 – Choose Appropriate Reference Sequence
  - Reduce the read count of each input read set (one for each platform) to the number of reads needed to cover 50% of the target viral genome (this figure is stored) via read lengths.
    - This is done to decrease runtime of the rest of this step.
  - Denovo-assemble the entire set of reads (after read count reduction) via CAP3.
  - Align the contigs to the chosen reference database, which contains several genome sequences of the target virus, and select the top hit with BLASTN.
  - This is the reference sequence.

Viral Assembly

- Step 3 – Reference Mapping and SNP Gathering.
  - Return the original read sets (full read counts).
  - 1 or 2 of the read sets are chosen for this step, based on their sequence platforms.
  - Which read sets are chosen follow this hierarchy, from first-choice to last:

1. 454 and Illumina
2. 454 and Iontorrent
3. 454 only
4. Illumina and Iontorrent
5. Illumina only
6. Iontorrent only
7. Sanger only

J. Craig Venter™ INSTITUTE

Viral Assembly

- For each read set, align to the reference sequence via BWA and gather all SNPs found via SAMTOOLS, BCFTOOLS, and VCFUTILS.
- Extract common SNPs found in both alignments via sdiff.
  - If only one read set was chosen (hierarchy choice 3, 5, 6, or 7), all SNPs are kept.

Viral Assembly

- Step 4 – Producing Final Viral Genome Sequence.
  - Modify the reference sequence with the SNPs found in Step 3 via delta2seq.
  - Align each read set to the modified reference via BWA.
  - Merge the sorted BAM of each BWA execution into one BAM via SAMTOOLS.
  - Extract the consensus sequence of the total alignment via SAMTOOLS, BCFTOOLS, and VCFUTILS.
  - This consensus sequence is the final, modified viral genome sequence.

VIGOR

- What is it?
  - VIral Genome Orf Reader (VIGOR).
  - A gene predictor program for small viral genomes.
  - VIGOR uses a similarity-based approach to detect ORFs by similarity searches against custom reference protein sequence databases.
  - Takes into account differences between the genomic structures of viral taxonomic groups.
  - Identifies frame-shifts, ribosomal slippage, RNA editing, stop codon read-through, overlapping genes, embedded genes, and mature peptide cleavage sites.
  - Genotyping capability for influenza and rotavirus is built into the program.

VIGOR

- VIGOR can be run from command-line via its Perl script wrapper.
- VIGOR is also publicly available as a webtool on JCVI's website.
- More information can be found here:
  - JCVI main VIGOR page and webtool: http://www.jcvi.org/vigor/index.php
  - Publication: http://www.biomedcentral.com/1471-2105/11/451

VIGOR

- For the VCR Pipeline, VIGOR is installed on the server and added as a Galaxy tool.
- The Galaxy tool has one input, a FASTA file of the sequence(s) the user wants annotated.
  - This is typically the output of Viral Assembly, but since VIGOR is a separate tool, the user can choose sequences from an outside source.

VIGOR

- Upon execution, the Galaxy tool executes the locally installed VIGOR on the FASTA file.
- Bundled with the VIGOR program itself, and installed alongside VIGOR during pipeline installation, are all the BLAST databases that VIGOR uses when annotating the input FASTA file.

VIGOR

- The Galaxy tool provides the following output files of VIGOR as output datasets:
  - ALN: CLUSTAL-W alignment file of genes found
  - CDS: FASTA file of features' nucleotide sequences
  - FS: logging data; this file is typically empty
  - PEP: FASTA file of features' peptide sequences
  - RPT: report of parameters used in run
  - TBL: tab-delimited file of discovered features
  - STATS: tab-delimited file coord/cov/ID data of features
- The Galaxy tool also uses VICVB (Venter Institute Cloud Viral Browser), a visualization module installed with VIGOR during pipeline installation, to produce an interactive JBrowse

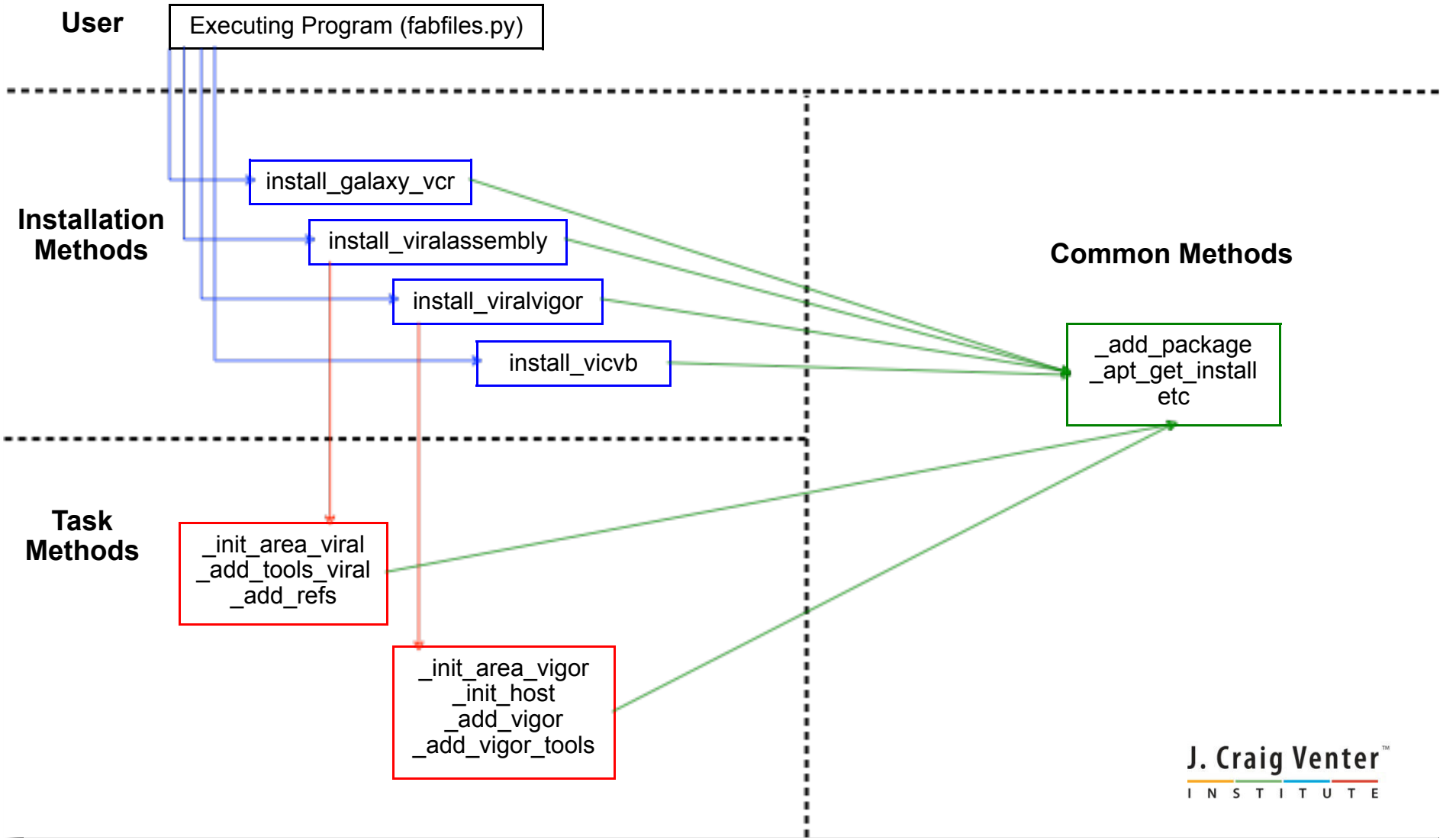
Pipeline Installation Script

- vcr.py
  - Stands for Viral Cloud Resource.
  - A python Fabric script for the following tasks:
    - Enables the environment to run the VCR pipeline.
      - Downloads and installs necessary dependencies (reference database files, 3rd-party tools, etc).
      - Installs the pipeline itself.
        - Adds the front-end files (XML and python) to the Galaxy instance (which must be present).
        - Installs the pipelines scripts on the box, and makes them accessible to the Galaxy front-end python scripts.

Pipeline Installation Script

- vcr.py has no main.
  - Like a library, it consists of functions that are used by a higher-level script (fabfiles.py) as appropriate.
  - A YAML config file (custom.yaml) allows outside scripts access to only a select few of these functions, which in turn make use of the vcr.py's other, deeper functions to carry out their tasks.
  - This allows the higher-level script to complete the entire installation of the pipeline via a few functions calls.
  - Keeps code compartmentalized and minimizes the number of locations to edit during modifications/updates.

Pipeline Installation Script



Pipeline Installation Script

- Installation Methods
  - install_galaxy_vcr
    - Installs the VCR Pipeline tools to the target Galaxy install. Downloads and properly places tools' XML/Python scripts and sets the necessary config files.
  - install_viralassembly
    - Installs the scripts and dependencies of Viral Assembly on the target VM.
  - install_viralvigor
    - Installs the scripts and dependencies of VIGOR on the target VM.
  - Install_vicvb
    - Installs the VICVB package.

J. Craig Venter™ INSTITUTE

Pipeline Installation Script

- Task Methods (install_viralassembly)
  - _initialize_area_viral
    - Defines paths and environment variables for Viral Assembly.
  - _add_tools_viral
    - Downloads and installs main script and dependencies of Viral Assembly in the locations set by _initialize_area_viral.
    - Also installs the bio-linux, allowing apt-get to access bioinformatic tools.
  - _add_refs
    - Downloads and installs the BLAST-able database files of the viral reference sequences in the locations set by _initialize_area_viral.

J. Craig Venter™ INSTITUTE

Pipeline Installation Script

- Task Methods (install_viralvigor)
  - _initialize_area_vigor
    - Defines paths and environment variables for VIGOR.
  - _initialize_host
    - Defines (and creates if not present) a location for VIGOR's by-product files.
  - _add_vigor
    - Downloads and installs the VIGOR package in the locations set by _initialize_area_vigor.
  - _add_tools_vigor
    - Downloads and installs the required programs and libraries, not included in the initial VIGOR package, in the locations set by _initialize_area_vigor.

Pipeline Installation Script

- Common Methods
  - _add_package
    - Downloads target package via wget to appropriate location and decompresses with corresponding program (tar, bz2, gzip, etc).
    - Used to download dependencies stored in Amazon S3 and FTP repositories.
  - _apt_get_install
    - Executes apt-get to install target tool (i.e. csh, gawk, bwa, samtools, etc)

Pipeline Installation Script

- vcr.py dependencies:
  - The target box must have Galaxy installed.
    - vcr.py assumes this is “/mnt/galaxyTools/galaxy-central”, but this can be changed by modifying the global variable galaxy_central.
  - Must have git, apt-get, and wget enabled.
    - This are used to install and download the needed programs and files from the various repositories.

Thanks too:

- The rest of the Team:
  - Ntino Krampis
  - Enis Afgan
  - Brad Chapman
  - Alex Richter
  - Andrey Tovchigrechko
- JCVI Viral Group
  - Tim Stockwell
  - Danny Katzel
  - Jeff Hoover

J. Craig Venter™
INSTITUTE