reportIT: IonTorrent Reporting Pipeline

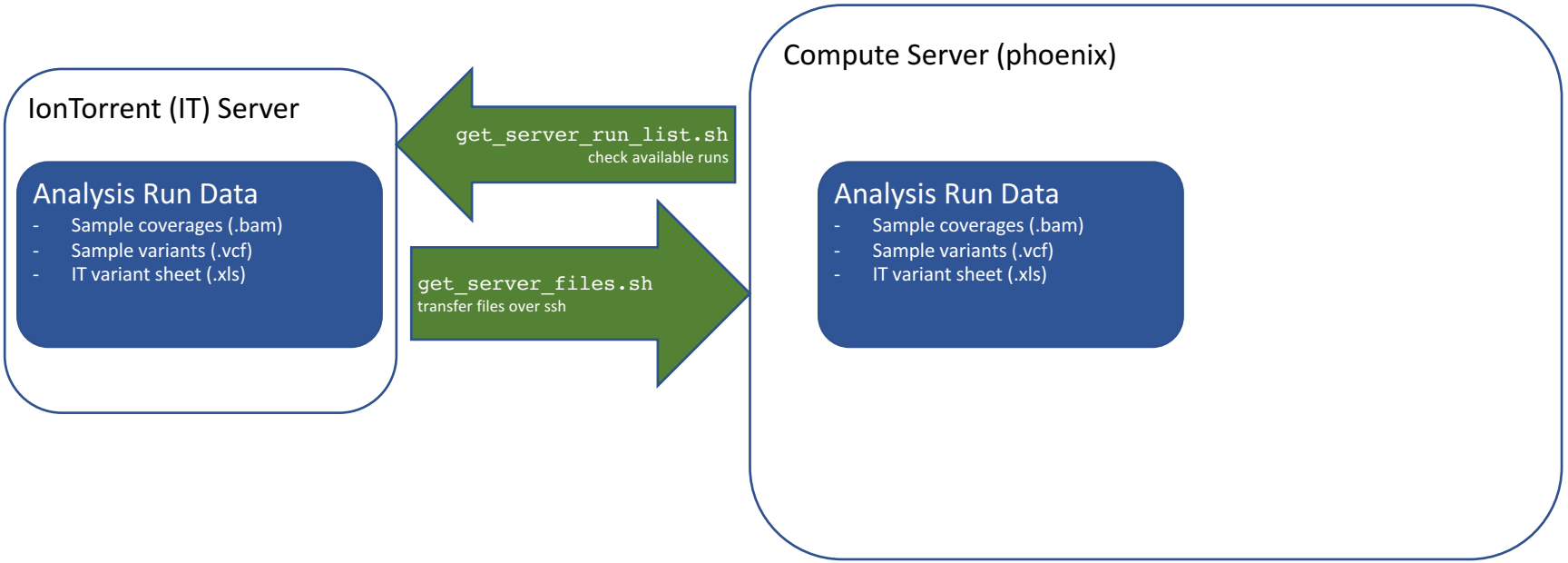
Stephen Kelly

NYU Langone Medical Center

March 1, 2017

https://github.com/stevekm/reportIT

Data Transfer



Analysis Pipeline Overview

Annotation

- `annotate_wrapper.sh / qsub_annotate_wrapper.sh`
  - `annotate_vcfs.sh`
    - split VCF entries with multiple variants (`bcftools`)
    - convert VCF to 'ANNOVAR input' format (`convert2annovar.pl`)
    - annotate with ANNOVAR (`table_annovar.pl`)
      - `-protocol refGene,cosmic68,clinvar_20150629,1000g2015aug_all -operation g,f,f,f`
    - convert 'ANNOVAR input' back into VCF format
    - query VCF for variant metadata fields (`bcftools`)
      - '%CHROM\t%POS\t%REF\t%ALT\t%QUAL\t%AF\t%FDP\t%FAO\t%STB\n'
    - convert VCF to TSV format (`vcf2tsv`)
  - `get_run_IDs.sh`
    - create index of Sample IDs, run Barcode IDs, and Run ID for downstream use
  - `merge_vcf_annotations_wrapper.sh`
    - `merge_vcf_annotations.py`
      - combine ANNOVAR variants & annotations with VCF metadata
      - process ANNOVAR output to create readable table format
      - filter for only 'canonical transcripts' amongst the variant annotations
      - add review field; Known/Unknown Significance
      - filter variant annotations based on quality criteria
        - Frequency > 0.05
        - Coverage > 250
        - exclude 'synonymous SNV'
        - Strand Bias < 0.8
        - MAF (1000g2015aug_all) < 0.01
      - save variant summary tables
    - add version control information to variant summary table
  - merge variant summary tables for all samples in the analysis run

Visualization & Reporting

- `IGV_report_wrapper.sh / qsub_report_wrapper.sh`
  - `IGV_snapshot_parser.sh`
    - `IGV_batchscript_generator.py`
      - create IGV batch script (.bat) based on variants from summary tables and coverage file (.bam), including NC sample coverage file if applicable
    - `run_IGV_batchscript.py`
      - run the IGV batch scripts (`IGV_2.3.81`) to create snapshots (.png) of sequencing reads at the locations where variants were found
  - `make_report_comments_wrapper.sh`
    - `make_report_comments.py`
      - match variants in the summary table with variants in the Weill Cornell IPM Knowledge Base
      - write IPMKB clinical comments into Markdown (.md) formatted file for each sample
  - `make_analysis_overview_report.sh`
    - `compile_RMD_report.R`
      - generate HTML overview report for all samples in the analysis, using IGV snapshots and variant summary tables
  - `make_full_report.sh`
    - `compile_RMD_report.R`
      - generate HTML report for each sample, using IGV snapshots, sample variant summary table, and clinical interpretations
- `IGV_report_wrapper-paired.sh / qsub_paired_report_wrapper.sh`
  - same as the report wrapper, but specifies ‘paired’ analyses that share control samples

Extras

- `run_samplesheet.py`
  - runs entire pipeline from a provided sample sheet
- `mail_analysis_report.sh`
  - aggregates analysis overview files & report, and emails them; uses parameters set in `mail_settings.sh`
- `dir_setup.sh`
  - configures a new repository with required binaries, reference data, and external directories
- `custom_bash_functions.sh, pipeline_functions.py, global_settings.sh`
  - locations, parameters, functions, and settings used throughout the pipeline
- `filter_criteria.json`
  - filter criteria used to filter the variant summary table