User Guide for DRAGEN COVID-19 Pipelines

The purpose of this document is to provide a quick user guide to analyze COVID-19 samples using the DRAGEN Platform on an on-site Server. Currently, there are 2 pipelines offered on DRAGEN for this purpose,

1. RNA Pathogen Detection pipelines
2. Metagenomics pipelines

RNA Pathogen Detection

Description of the pipeline

This pipeline aligns reads to a reference combining the human genome with a collection of viral sequences. The RNA-seq aligner is used, supporting spliced alignments for human reads in the sample. Presence or absence of each virus in the reference is detected based on reads aligning to the virus sequence, specifically using the percentage of the viral genome covered by unique alignments.

The alignments can then be used to perform variant calling and generate a consensus fasta with the specific virus strain sequence.

Inputs

- Input reads (fastq)
- Reference genome (DRAGEN hash-table)
    - Including Human genome, Sequences / contigs for all targeted viruses
    - Anchored (RNA) hash-table for spliced alignment
- Virus target regions (BED)
    - Identifies virus contigs in the reference hash-table
    - Format (tab-separated): <ContigName> <0> <ContigLength> <VirusName>

Outputs

- Alignments (BAM) – DRAGEN map/align
- Per-contig coverage (qc-coverage-reports) – DRAGEN map/align
- Virus detection report (csv) – python script
- Variant calls (VCF) - DRAGEN VC

Example Command lines

DRAGEN Map/Align:

(Generates alignments and per-contig coverage metrics)

/opt/edico/bin/dragen --bin_memory 42949672960 --enable-map-align true --enable-rna true--enable-duplicate-marking true --dupmark-version hash --enable-kmer true --kmer-dropLowComplexity true --kmer-kmerFASTA /staging/files/fasta/virus/SFS_SARS-CoV-2.fa --kmer-reverseComplement true --output-format BAM --qc-coverage-tag-1 pathogen --qc-coverage-region-1 /staging/files/bed/virus/respiratory-panel.bed --qc-coverage-reports-1 overall_mean_cov hist contig_mean_cov cov_report full_res --vc-target-bed /staging/files/bed/virus/respiratory-panel.bed --output-directory /data/output/appresults/168316154/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2_229053826 --output-file-prefix SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2 --enable-metrics-json true --tumor-fastq-list /data/scratch/fastq_sheet.csv --tumor-fastq-list-sample-id SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2 --ref-dir /data/scratch/hg38_respiratory_panel.v8

DRAGEN Variant Calling:

(Generates VCF from alignments)

/opt/edico/bin/dragen --bin_memory 42949672960 --output-directory /data/output/appresults/168316154/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2_229053826/vc --output-file-prefix SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2 --enable-map-align false --enable-variant-caller true --tumor-bam-input /data/output/appresults/168316154/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2_229053826/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2_tumor.bam --enable-metrics-json true --vc-ignore-cigar-skip-reads true --ploidy 1 --ref-dir /data/scratch/hg38_respiratory_panel.v8

Virus Coverage:

(Uses generic coverage metrics from map/align output to report coverage for each virus)

/staging/post_process/virus_coverage.py /staging/files/bed/virus/respiratory-panel.bed /data/output/appresults/168316154/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2_229053826/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2.pathogen_cov_report.bed /data/output/appresults/168316154/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2_229053826/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2.pathogen_full_res.bed > /data/output/appresults/168316154/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2_229053826/SRR11247076_RNA-Seq-of-Severe-acute-respiratory-syndrome-coronavirus-2.pathogen-coverage-report.tsv

Optional Command line options

None

Metagenomics Pipeline

Description of the pipeline

DRAGEN-metageomics utilize the DRAGEN map-align engine in de-hosting and the core logics from Kraken2 in read taxon classification.

Inputs

The user must supply a FASTQ through the standard DRAGEN interface, and the 3 files with names taxo.k2d, opts.k2d and index.k2d generated from Kraken2 index build. When de-hosting is on the user must supply a host DRAGEN reference index. This reference index is a standard DRAGEN map align index. The following sections show the command line usages when using 1.) de-hosting and 2.) without de-hosting.

Outputs

The DRAGEN metagenomics output consist of file names in standard DRAGEN format {output_file_prefix}_{file_postfix}. Here are output files specific to DRAGEN metagenomics:

1. a file with postfix _microbe-classification.tsv, which consist of the taxon classification of each read from Kraken2. Here is the file format of this file (copied from Kraken2 website, https://ccb.jhu.edu/software/kraken2/index.shtml?t=manual#standard-kraken-output-format):
    a. "C"/"U": a one letter code indicating that the sequence was either classified or unclassified.
    b. The sequence ID, obtained from the FASTA/FASTQ header.
    c. The taxonomy ID Kraken 2 used to label the sequence; this is 0 if the sequence is unclassified.
    d. The length of the sequence in bp. In the case of paired read data, this will be a string containing the lengths of the two sequences in bp, separated by a pipe character, e.g. "98|94".
    e. A space-delimited list indicating the LCA mapping of each k-mer in the sequence(s).

2. a file with postfix _microbe-classification-report.tsv, which consist of taxon relative abundance from Kraken2. Here is the file format of this file (copied from Kraken2 website, https://ccb.jhu.edu/software/kraken2/index.shtml?t=manual#sample-report-output-format):
    a. Kraken 2's standard sample report format is tab-delimited with one line per taxon. The fields of the output, from left-to-right, are as follows:
    b. Percentage of fragments covered by the clade rooted at this taxon
    c. Number of fragments covered by the clade rooted at this taxon
    d. Number of fragments assigned directly to this taxon
    e. A rank code, indicating (U)nclassified, (R)oot, (D)omain, (K)ingdom, (P)hylum, (C)lass, (O)rder, (F)amily, (G)enus, or (S)pecies.
    f. NCBI taxonomic ID number
    g. Indented scientific name

3. a file with postfix _microbe-classification_metrics.csv: which shows the number reads filtered after de-hosting and the number of classified reads.

Example Command lines

With de-hosting

```
dragen -r
/illumina/scratch/DRAGEN/data/vault/reference_genomes/Hsapiens/GRCh38_full_plus_decoy_hla/DRAGEN/8 -1 /home/bytsui/Code/metaphlan_test/kraken2/my_kraken2/test.fastq --metagenome-taxonomy
/illumina/scratch/DRAGEN/users/bytsui/metagenome_data/minikraken_8GB_20200312/taxo.k2d
--metagenome-options
/illumina/scratch/DRAGEN/users/bytsui/metagenome_data/minikraken_8GB_20200312/opts.k2d
--metagenome-index
/illumina/scratch/DRAGEN/users/bytsui/metagenome_data/minikraken_8GB_20200312/hash.k2d
--output-directory /staging/bytsui/tmp_out --output-file-prefix test_sample --RGID RG --RGSM SM
--Aligner.aln-min-score 50 --enable-sorting false
```

Without de-hosting

```
dragen -1 /home/bytsui/Code/metaphlan_test/kraken2/my_kraken2/test.fastq -r
/illumina/scratch/DRAGEN/data/vault/reference_genomes/Hsapiens/GRCh38_full_plus_decoy_hla/DRAGEN/8 --metagenome-taxonomy
/illumina/scratch/DRAGEN/users/bytsui/metagenome_data/minikraken_8GB_20200312/taxo.k2d --metagenome-options
/illumina/scratch/DRAGEN/users/bytsui/metagenome_data/minikraken_8GB_20200312/opts.k2d --metagenome-index
```

/illumina/scratch/DRAGEN/users/bytsui/metagenome_data/minikraken_8GB_20200312/hash.k2d --output-directory /staging/bytsui/tmp_out --output-file-prefix test_sample --RGID RG --RGSM SM --Aligner.aln-min-score 10000 --enable-sorting false

Optional Command line options

--enable-metagenome: set to true to enable DRAGEN metagenomic pipeline

--metagenome-taxonomy: points to the taxo.k2d file from Kraken2 index build.

--metagenome-options: points to the opts.k2d file from Kraken2 index build.

--metagenome-index: points to the hash.k2d file from Kraken2 index build.

--output-directory: standard DRAGEN output directory, which will contain the output for DRAGEN metagenomics also.

-r: points to the host reference, which will be used in de- hosting

--Aligner.aln-min-score: Please adjust the DRAGEN Map Alignment threshold --Aligner.aln-min-score between 0 and your FASTQ read length for de-hosting sensitivity tuning. To disable de-hosting please set --Aligner.aln-min-score to a very high number well beyond the read length, e.g. 10,000.

-- enable-sorting: Please note that you can disable sorting with --enable-sorting false to reduce runtime by 6X, depending on the portion of host reads.