I have a slew of scripts posted to GitHub under EDICO\_SCRIPTS that I have been playing with (some not worth it and one functioning from fastq\_pipeline outlined below).

Documentation

* Dragen Documentation can be found here (workflow demo scripts, user guide, etc.): **/isilon/sequencing/Dragen\_Documentation/**
* WHERE TO FIND RELEASES AND ALL THAT JAZZ: [https://dragen.edicogenome.com](https://dragen.edicogenome.com/)
  + Double check your login -- When I logged in with that username and password sent to you I was not able to view releases and user guide documentation for some reason
    - -- e-mail title was "Welcome to the Edico Genome Customer Portal"
* Primary person of contact: Shyamal Mehtalia <shyamal@edicogenome.com>
  + Quick to respond (usually within a few hours or so)
* Directories of importance:
  + /isilon/sequencing/VITO/NEW\_GIT\_REPO/EDICO\_SCRIPTS/ -- Where all the scripts I've messed with live.  Trying to update them before I depart to make running multiple samples through dragen (gvcf,vcf, ms\_calling) one step submission process
  + /isilon/sequencing/Seq\_Proj/EDICO\_TEST/ -- Single sample test where the comparison to bina was generated
  + /isilon/sequencing/Seq\_Proj/EDICO\_TRIO\_TEST/ -- Trio from Macrogen used through dragen and data generated for the ppt slides
  + /isilon/sequencing/Seq\_Proj/edico\_vm\_test/ -- Test run for the trio again.  Just working out the kinks of submitting from the VM.
* Server names
  + Dragen1: dragen1.cidr.jhmi.edu
  + Dragen2: dragen2.cidr.jhmi.edu
  + Dragenapi: dragenAPI.cidr.jhmi.edu -- Specific system requirements were needed to install some edico software/slurm setup to communicate appropriately
    - RH 6.8 (Santiago) -- Tried to use 7 at first, but the person installing from edico stated the incompatibility
    - 2 Processors
    - 8GB memory

Some Things to Note

**\*\*Everthing must be submitted as root regardless of being on the server itself or from the VM\*\***

**\*\*As Root you must also set these limits:**

**# ulimit -n 65535**

**# ulimit -s 10240**

**# ulimit –u 16384**

**\*\*If you cancel a job on the dragen nodes prematurely, you must run dragen\_reset on the node manually.  If the job dies on it's own, the dragen\_reset command is ran automatically.\*\***

Scripts that runs on dragen (directly or from the Dragenapi VM – doesn’t matter)

Standard pipeline I ran for samples:

**from\_fastq\_project\_submission.sh** - This would generate both job submissions to create the gvcf/bam and vcf for each sample found in the sample sheet. This will also generate a joint\_calling job submission that will run once the appropriately generated gvcfs have been generated for all samples within the sample sheet.

      -$1=standard sample sheet

-$2=Dir of fastq files -- files need to be named conventional RGPU\_1 RGPU\_2 to generate the appropriate fastq\_list.csv

      -$3=OutDir - full path of project

* **from\_fastq\_vcf.sh** (submitted by from\_fastq\_project\_submission.sh)
  + Produces a bam file, hard-filtered vcf, and vcf
* **from\_fastq\_gvcf.sh** (submitted by from\_fastq\_project\_submission.sh)
  + Produces a gvcf
* **joint\_calling.sh** (submitted by from\_fastq\_project\_submission.sh)
  + Script parses the GVCF folder where all the gvcf files created from "from\_fastq\_gvcf.sh" script are moved over to after being made

Stuff that runs on sge that needs to be submitted manually post dragen

Run "**/isilon/sequencing/VITO/NEW\_GIT\_REPO/EDICO\_SCRIPTS/Post\_Single\_Sample\_Processing** **/SCRIPTS/SUBMITTER.QC.LUMPY\_DRAGEN.sh**" after the single sample portion is completed, for single sample stats

* $1=Sample\_Sheet Can use the regular style sample sheet.  Before it was calling for a bina style sample sheet.  Made slight alterations (not a complete overhaul) to just use the standard samplesheet we use for everything else.
* $2=Project
* Removed concordance (time consumption) and estimate library (duplicates not marked)

Next for a SS\_QC\_REPORT, run "**/isilon/sequencing/VITO/NEW\_GIT\_REPO/EDICO\_SCRIPTS/Post\_Single\_Sample\_Processing/SCRIPTS/QC\_REPORT\_WHOLE\_GENOME\_SINGLE.sh**"

* $1=Core\_Path
* $2=Project

Run "**/isilon/sequencing/VITO/NEW\_GIT\_REPO/EDICO\_SCRIPTS/Post\_Multi\_Sample\_Processing/SCRIPTS/00\_SUBMITTER.MULTI.SAMPLE.REPORTS.sh**" after joint\_calling.sh is completed

* $1=Project
* $2=Sample\_Sheet
* Scripts will run GT\_Refinement on MS\_VQSR vcf from dragen through vcf extractions
* Removed concordance (time consumption)

Next for a MS\_QC\_REPORT, run "**/isilon/sequencing/VITO/NEW\_GIT\_REPO/EDICO\_SCRIPTS/Post\_Multi\_Sample\_Processing/SCRIPTS/01\_QC\_REPORT\_WHOLE\_GENOME\_MULTI.sh**"

* $1=Project