PharmaSUG 2024 - Paper PO- 440

The SAS Genome – Genetic Sequencing

Oliver Lu, Eurofins Viracor BioPharma Services, Inc.
Katie Watson, Eurofins Viracor BioPharma Services, Inc.

ABSTRACT

After COVID-19 caused a global shake-up, a growing number of pharmaceutical companies began to place increasing emphasis on genetic sequencing alongside their requested sample testing. Sequencing data outputs can often contain a gargantuan amount of data in various formats, and there may be customization requests to contend with as well. As a programmer you may need to convert the files and organize the data into a comprehensive SAS dataset or report.

INTRODUCTION

Genomic sequencing techniques are continually improving and advancing as their importance is recognized in deciphering viral strain traits within the gigabases of genetic anatomy. With genetic sequencing popularity climbing alongside viral testing, you will likely encounter some form of SAS programming request pertaining to sequencing data. Sequencing pipeline setup can vary depending on the laboratory methodology and the Sponsor's area of interest. SAS programming can assist in providing further flexibility and customization to meet Sponsor needs. From compiling several files of incremental batch data into cumulative reports to creating demographic key files that complement the sequencing output files, SAS programming innovations can be used to organize the data in a meaningful way. We will review the advantages of creating a corresponding key file to complement sequencing output files, useful tips to help you compile the data, and a few of the various sequencing types/categories you could potentially encounter.

SEQUENCING DECODER RINGS AND MUTANTS

Depending on the structure of your LIMS system and the output of the sequencing pipeline, it may be helpful to provide the sponsor with a key file that includes identifying demographic information to align back with the samples.

DEMOGRAPHIC KEY FILES FOR SUBJECT DATA IDENTIFICATON

Demographic key files are provided to sponsors to assist with sample identification. They can be used to match the sequencing output files back to the corresponding Subject ID/Sponsor Accession ID. A key file can also be helpful because it allows you to include test and visit mapping in the file that otherwise may not be possible due to the configuration of the pipeline setup.

Figure 1 is an example of a sequencing key table to assist with matching up demographic information with corresponding samples.

#	Variable	Label	Type	Len	Format	Notes
1	STUDYID	Study Identifier	Char	25		
2	SUBJID	Subject Identifier for the Study	Char	15		
3	VISIT	Visit Name	Char	20		
4	VISITNUM	Visit Number	Char	10		
5	LBDT	Date of Specimen Collection	Num	8	DATE9.	DDMMMYYYY
6	LBTM	Hour of Specimen Collection	Num	8	TIME5.	HH:MM
7	LBTESTCD	Lab Test or Examination Short Name	Char	10		
8	LBTEST	Lab Test or Examination Name	Char	60		
9	LBORRES	Result or Finding in Original Units	Char	50		
10	LBORRESU	Original Units	Char	10		
11	LBREFID	Specimen ID	Char	30		
12	ACCESNUM	Sponsor Accession Number	Char	15		
13	LBCOM	Comment	Char	200		

Figure 1. WGS Key File variable table for programming in SAS 9.4

S-GENE FILE OUTPUTS

Identifying and listing observed spike gene mutations is an example of additional secondary analysis sequencing samples may undergo. Sponsors may be interested in the number of spike mutations related to potential S-Gene variants. The following table and programming setup helps to pull in files to generate a cumulative S-Gene output file for review.

Figure 2 is an example of a S-Gene table to provide potential spike mutations that may be observed. SPIKE_MUTATION_n is representative of any additional spike mutations that may be included.

#	Variable	Label	Type	Len	Notes
1	STUDYID	Study Identifier	Char	40	
2	SUBJECT_ID	Subject ID	Char	15	
3	COLLECTION_DATE	Date of Sample Collection	Char	10	
4	SPONSOR_BARCODE	Sponsor Barcode	Char	20	
5	LAB_ACCESSION	Lab Accession	Char	20	
6	STATUS	Status	Char	20	
7	LINEAGE	Lineage	Char	20	
8	PANGO_VERSION	Pango Version	Char	20	
9	SPIKE_MUTATION_1	Spike Mutation 1	Char	15	
10	SPIKE_MUTATION_2	Spike Mutation 2	Char	15	
11	SPIKE_MUTATION_3	Spike Mutation 3	Char	15	
12	SPIKE_MUTATION_4	Spike Mutation 4	Char	15	
13	SPIKE_MUTATION_5	Spike Mutation 5	Char	15	
14	SPIKE_MUTATION_6	Spike Mutation 6	Char	15	
15	SPIKE_MUTATION_7	Spike Mutation 7	Char	15	
29	SPIKE_MUTATION_n	Spike Mutation n	Char	15	

Figure 2. S-Gene table for programming in SAS 9.4

GENETIC IMPORTS AND REPORT MATCHMAKING

Sponsors may have specific formatting requirements for sequencing data, which can require customization. This can create challenges manually importing many files into SAS. However, there are occasions when directory file importing can be used to expediate the process. Instead of providing files from each individual sequencing pipeline batch run, it can sometimes be helpful to compile corresponding data outputs from each batch run into one aggregate file to simplify data review. In these instances, rather than importing each file one at a time, it can save time to place all files in a directory and directly import all files.

The SAS programming code below was applied to a study that required the SAS import of a series of CSV files generated by a sequencing output pipeline created by our laboratory scientists. This was an opportunity for SAS programming intervention and support, and an instance where SAS programming could be integrated to assist with generating and processing sponsor customized Next-Generation Sequencing [NGS] table exports. This data would later be submitted as part of a final report for FDA review.

As sequencing samples are tested and run via a structured sequencing output reporting pipeline, multiple CSV files can potentially be generated from a batch. The specific batch data will then need to be combined into a cumulative dataset for reporting purposes to the FDA or even processed for individual reports. In this case, the following IMPORT procedure can be used to import each individual *.csv stored within a designated directory (SAS Help Center 2024).

Prior to applying the macro variable codes to import the CSV files, it is important to ensure the output CSV files formatting/contents/table(s) are all consistently structured in the same way. This will prevent any potential issues importing the CSV data and allow for a more meaningful character-length read as the SAS program determines a max limit for each column variable depending on the imported records/rows. However, additional programming can be added to establish a manual character length per column/variable. When working with your laboratory scientists, it can be beneficial to inquire if unique sample ID numbers/demographics can be included in the CSV file naming convention or within the CSV file contents itself. If both are available for reference, this is even better as it will allow further options for handling the data, adding additional programming flexibility.

The macro code utilizes set macro definitions and parameters to assign values that allow the SAS user to access a designated directory stored in the macro variable ‘dir’. Once the designated folder containing all the individual sequencing output CSV files has been established, the macro variable steps can be initiated to attempt to open/access the directory, thereby checking to see if the directory can be successfully opened using the code. Later steps in the macro variable coding involve IF/THEN statements that verify the specified CSV extension matches the file type being read. As each of the CSV files are being read in, the data set named ‘DSNx’ is being created and will come in handy later when combining the individual CSV file datasets (SAS Help Center 2024).

SAS programming code for import of multiple CSV files within a directory:

```
%macro drive(dir,ext);
      %local cnt filrf rc did memcnt name;

      %let cnt=0;
      %let filrf=mydir;

      %let rc=%sysfunc(filename(filrf,&dir));
      %let did=%sysfunc(dopen(&filrf));

      %let EFI_ALLCHARS=YES;

      %if &did ne 0 %then %do;
            %let memcnt=%sysfunc(dnum(&did));
            %do i=1 %to &memcnt;
            %let name=%qscan(%qsysfunc(dread(&did,&i)),-1,.);

                  %let name2=%qscan(%qsysfunc(dread(&did,&i)),-2,.);

  %if %qupcase(%qsysfunc(dread(&did,&i))) ne %qupcase(&name) %then %do;
      %if %superq(ext) = %superq(name) %then %do;
         %let cnt=%eval(&cnt+1);
         %put %qsysfunc(dread(&did,&i));

proc import datafile="&dir\%qsysfunc(dread(&did,&i))"
      out=dsn&cnt(label="&name2")
                  dbms=csv replace;
                  guessingrows=max;
run;

DATA DSN&CNT;
      SET DSN&CNT;
      LENGTH FILE_ID $27;
      RETAIN FILE_ID "&CNT._&name2"; /*USED TO IDENTIFY THE FILE
IMPORTED*/
RUN;
/*%let EFI_ALLCHARS=NO;*/ /***NOTE: By commenting out this segment, all
 proc import csv files will be read in as char***/
                  %end;
              %end;
          %end;
      %end;
%else %put &dir cannot be opened.;
%let rc=%sysfunc(dclose(&did));
%mend drive;

%drive(S:\BioPharma\BioPharma SAS Studies\2 -
Sequencing\NGS\SPONSOR\RSV_NGS_ALL\RSV_NGS_IMPORT,csv)
```

The following macro variable coding lists where the directory containing the CSV files is located, while the second section after the comma describes the type of file extension being imported (SAS Help Center 2024):

```
%drive(S:\BioPharma\BioPharma SAS Studies\2 -
Sequencing\NGS\SPONSOR\RSV_NGS_ALL\RSV_NGS_IMPORT,csv)
```

The beauty of this macro variable import is the inclusion of the PROC IMPORT function which functions like a DNA polymerase as it reads/imports the CSV files. The PROC IMPORT function can scan either the first 20 rows of the imported CSV file, or it can be modified using ‘guessingrows=max’ to instruct the import procedure to scan all the input data rows instead of merely the first 20 (SAS Help Center 2024). By scanning more than 20 rows, this will help in lowering the risk of data values being truncated:

```
proc import datafile="&dir\%qsysfunc(dread(&did,&i))"
out=dsn&cnt(label="&name2")
                  dbms=csv replace;
                  guessingrows=max;
run;
```

Using the information stored from the creation of data sets named ‘DNSx’, with the ‘x’ keeping a sequential count of each of the CSV files stored in the designated directory, you can later access the active SAS session to recall information from the SAS dictionary tables for ‘MEMNAME’ from the table ‘dictionary.members’. A cumulative file can then be generated from all the imported CSV files as they are combined into one file based on the assigned ‘DSNx’ SAS table naming system. The variable column ‘FILE_ID’ will also be retained in its respective column and can be useful, depending on the information that it provides and its ability to differentiate each separate file. Table 4 provides an example of how you can use the SAS dictionary table to create your cumulative sequencing dataset file and manually set specific column character lengths that are of interest for your final sequencing report output file.

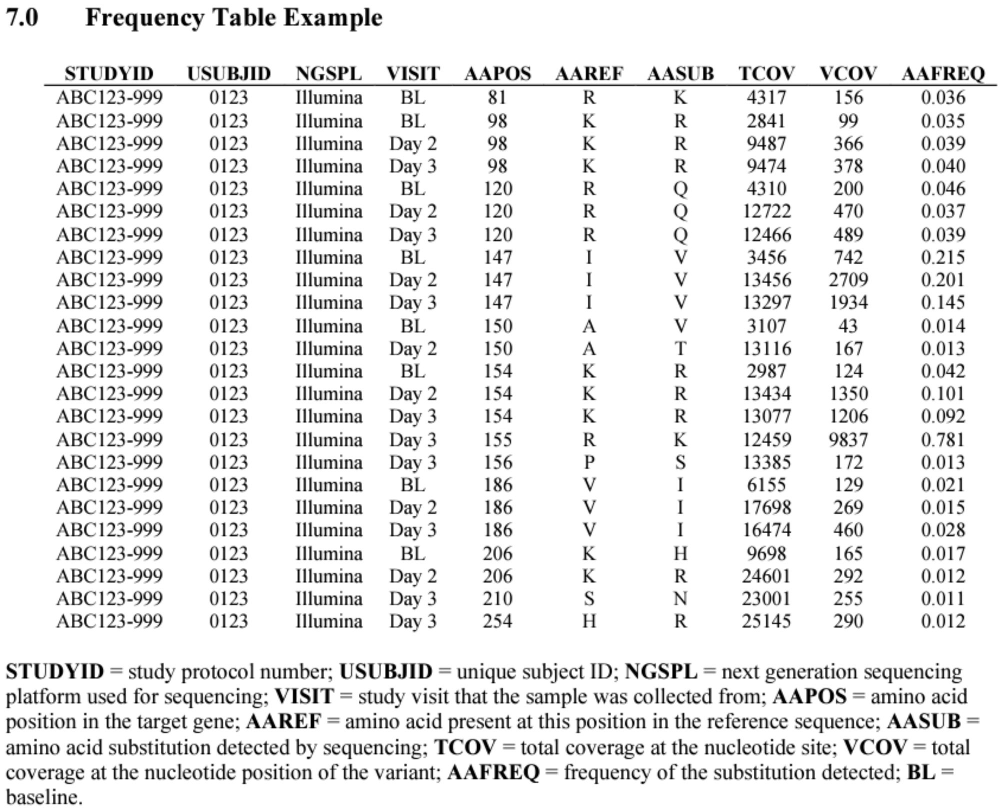
SAS programming code for combining multiple imported CSV files:

```
PROC SQL NOPRINT;
      SELECT DISTINCT MEMNAME INTO :DSNLIST SEPARATED BY ' '
            FROM DICTIONARY.TABLES
                  WHERE LIBNAME='WORK' AND MEMNAME LIKE "DSN%";
QUIT;
DATA OUTPUT.RSV_NGS_&CVID._MULTI;
      LENGTH      Region                                    $15
                  Reference                                 $5
                  Allele                                    $5
                  Zygosity                                  $20
                  Count                                     $15
                  Coverage                                  $15
                  Frequency                                 $15
                  Reverse_read_count                        $15
                  Reverse_read_coverage                     $15
                  Read_count                                $15
                  Read_coverage                             $15
                  Coding_region_change                      $30
                  Coding_region_change_in_longest           $30
                  Other_variants_within_codon               $5
            Amino_acid_change                               $50
            Mapped_reads____                                $30 ;
      SET &DSNLIST;
RUN;
```

A generic and more detailed program description can be found at the SAS Help Center at the website listed under the *References* section.

For the submission to the FDA for Next-Generation Sequencing data outputs, Table 5 provides an example from an FDA submission guidance PDF, detailing specific information that is of interest (FDA 2019). This was a similar table provided as a reference point in structuring the final report layout and structure to generate a Frequency Variant Report Table.

Figure 3 Example of FDA submission guidance on next generation sequencing data (FDA 2019)



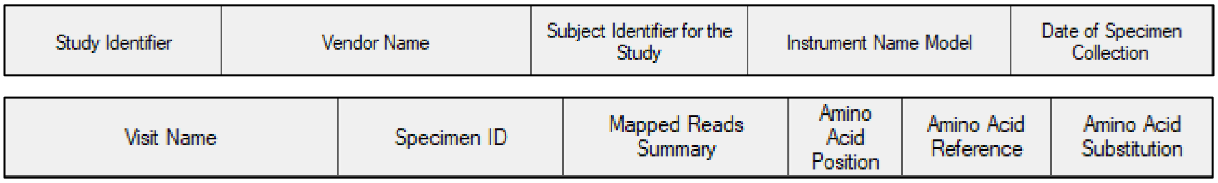
7.0 Frequency Table Example

STUDYID	USUBJID	NGSPL	VISIT	AAPOS	AAREF	AASUB	TCOV	VCOV	AAFREQ
ABC123-999	0123	Illumina	BL	81	R	K	4317	156	0.036
ABC123-999	0123	Illumina	BL	98	K	R	2841	99	0.035
ABC123-999	0123	Illumina	Day 2	98	K	R	9487	366	0.039
ABC123-999	0123	Illumina	Day 3	98	K	R	9474	378	0.040
ABC123-999	0123	Illumina	BL	120	R	Q	4310	200	0.046
ABC123-999	0123	Illumina	Day 2	120	R	Q	12722	470	0.037
ABC123-999	0123	Illumina	Day 3	120	R	Q	12466	489	0.039
ABC123-999	0123	Illumina	BL	147	I	V	3456	742	0.215
ABC123-999	0123	Illumina	Day 2	147	I	V	13456	2709	0.201
ABC123-999	0123	Illumina	Day 3	147	I	V	13297	1934	0.145
ABC123-999	0123	Illumina	BL	150	A	V	3107	43	0.014
ABC123-999	0123	Illumina	Day 2	150	A	T	13116	167	0.013
ABC123-999	0123	Illumina	BL	154	K	R	2987	124	0.042
ABC123-999	0123	Illumina	Day 2	154	K	R	13434	1350	0.101
ABC123-999	0123	Illumina	Day 3	154	K	R	13077	1206	0.092
ABC123-999	0123	Illumina	Day 3	155	R	K	12459	9837	0.781
ABC123-999	0123	Illumina	Day 3	156	P	S	13385	172	0.013
ABC123-999	0123	Illumina	BL	186	V	I	6155	129	0.021
ABC123-999	0123	Illumina	Day 2	186	V	I	17698	269	0.015
ABC123-999	0123	Illumina	Day 3	186	V	I	16474	460	0.028
ABC123-999	0123	Illumina	BL	206	K	H	9698	165	0.017
ABC123-999	0123	Illumina	Day 2	206	K	R	24601	292	0.012
ABC123-999	0123	Illumina	Day 3	210	S	N	23001	255	0.011
ABC123-999	0123	Illumina	Day 3	254	H	R	25145	290	0.012

STUDYID = study protocol number; **USUBJID** = unique subject ID; **NGSPL** = next generation sequencing platform used for sequencing; **VISIT** = study visit that the sample was collected from; **AAPOS** = amino acid position in the target gene; **AAREF** = amino acid present at this position in the reference sequence; **AASUB** = amino acid substitution detected by sequencing; **TCOV** = total coverage at the nucleotide site; **VCOV** = total coverage at the nucleotide position of the variant; **AAFREQ** = frequency of the substitution detected; **BL** = baseline.

Figure 3. FDA submission guidance example for NGS data output

Figure 4 Example of NGS output SAS header labels for final reporting



Study Identifier	Vendor Name	Subject Identifier for the Study	Instrument Name Model	Date of Specimen Collection

Visit Name	Specimen ID	Mapped Reads Summary	Amino Acid Position	Amino Acid Reference	Amino Acid Substitution

Figure 4. SAS header labels for final NGS output reporting example

SEQUENCING CATEGORIES

Below here is a list of various gene sequencing techniques performed by laboratories that you may encounter as a programmer:

- WGS: Whole Genome Sequencing
- NGS: Next-Generation Sequencing
- SGS: Sanger Sequencing
- WES: Whole Exome Sequencing

Types of data output files that may appear in the sequencing output files:

SGS Output Files:

Main Folder Name	Sub Folder Name[s] And Pathway	Type of File	Description of Document	Document Name	Report Status
X/Y Folder	**ABI FILE FOLDER**	***.ab1**	Applied Biosystems Sequencer Data File	[Sponsor Accession]-[X/Y]-1_[Sanger Sequencing Primer]**.ab1**	Conditional
	FASTA FILE FOLDER > [X/Y] SUBTYPE	***.fsta**	Consensus Sequence	[Sponsor Accession]-[X/Y]-1**.fsta**	Conditional
	STATISTICS REPORT FOLDER	***.pdf**	Statistics Report	[Sponsor Accession]-[X/Y]-1_Statistics_Report**.pdf**	Conditional
	VARIANT REPORT FOLDER		AA Variants Report	[Sponsor Accession]-[X/Y]-1_AA_Variants_Report**.pdf**	Conditional
Results Folder	**-**	***.xlsx**	Gene Sequencing Table	RSV Cumulative Gene Sequencing**.xlsx** RSV[X/Y] Gene Sequencing**.xlsx**	Conditional

Figure 5. Examples of SGS Output Files

WGS Output Files:

Type of File	Description of Document	Document Name	Report Status
*html	Summary Report	US-<Subject ID>_Analysis_Report.html	N/A
*txt	Lineage Report	US-<Subject ID>_pangolin_lineage_classification.txt	N/A
*fasta	Sample fasta	US-<Subject ID> _<accession>.consensus.fasta	Conditional
*html	Sample Surveillance Report	US-<Subject ID>_<accession> _Surveillance_Report.html	Conditional

Figure 6. Examples of WGS Output Files

NGS Output Files:

Type of File	Description of Document	Document Name
*.txt [fasta]	Consensus	<Run#>_<RSV Type >_NGS_Consensus.txt
*.xlsx	Variable Table and Mapping Coverages	<accession> (Sample report).xlsx
*.xlsx	Mapping Coverage Table	<Run#>_<RSV Type>_NGS_Mapping Coverage Table.xlsx
*.jpg	Coverage Graph	<accession> mapping (coverage graph).jpg
*.fastq.gz	Raw Data	Files ending with suffix ".fastq.gz" <accession>-<X1/X2>_<S##>_<L###>_<Y1/Y2>_<###>.fastq.gz
*.xlsx	Trim Report	RSVC: <YYMMDD[Run#]>_<RSVC[RSV Type]>_NGS_Trim Report.xlsx RSVE: RSVE_<Run#>_<DDMMMYY>_Trim Report.xlsx
*.xlsx	List of Samples Ran at Low Volume	Lowvolume.xlsx

Figure 7. Examples of NGS Output Files

WES Output Files:

Main Folder[s]	Subdirectory Folder[s] and File[s]
*Results	**{run_folder_id}** ▪ **Logs_Intermediates** • **CollapsedReads -** Subfolders per sample ID containing the aligned BAM and index files; CollapsedReads output logs • **CombinedVariantOutput** • **DnaAlignment -** Subfolders per sample ID containing the aligned BAM and index files; DnaAlignment output logs • **DnaFastqValidation** • **DnaQCMetrics -** Subfolders per sample ID containing the aligned, collapsed, and stitched metrics JSON. Files; DnaQCMetrics output logs • **DnaRealignment** - Subfolders per sample ID containing the realigned BAM and index files; DnaRealignment output logs • **FastqDownsample -** Subfolders per sample ID containing FASTQ files; FastqGeneration Output Logs • **FastqGeneration** - Subfolders per sample ID containing FASTQ files; FastqGeneration Output Logs • **FusionCalling -** Subfolders per sample ID containing the genome VCF file; FusionCaller output logs • **PhasedVariants -** Several folders with this name may exist in the output folder structure; Subfolders per sample ID containing the phased variant metrics JSON; Psara and Scylla output logs • **ResourceVerification** • **RnaAlignment** - Subfolders per sample ID containing the aligned BAM and index files; RnaAlignment output logs • **RnaFusionFiltering** • **RnaFusionMerge -** Subfolders per sample ID containing the CSV file listing all the fusions; RnaFusionMerge output logs • **RnaMarkDuplicates -** Subfolders per sample ID containing the marked, aligned BAM and index files; RnaMarkDuplicate output logs

  - **RnaQCMetrics -** Subfolders per sample ID containing the aligned, collapsed, and stitched metrics JSON files; RnaQCMetrics output logs
  - **RnaSpliceVariantCalling -** Subfolders per sample ID containing the splice variants VCF and fusion TSV files; dsdm JSON
  - **RunQc** [RunQC Metrics JSON file; RunQC Output logs]
  - **SampleAnalysisResults**
  - **SamplesheetValidation**
  - **SmallVariantFilter -** Subfolders per sample ID containing the error rate tables; SmallVariantFilter output logs
  - **StitchedRealigned -** Subfolders per sample ID containing the stitched, realigned BAM and index files. The StitchedRealigned BAM is the final output BAM for DNA; StitchedRealigned output logs
  - **Tmb -** Subfolders per sample ID containing the TMB metrics JSON; TMB output logs
  - **TrimFastq - Subfolders per sample ID containing FASTQ files; dsdm JSON**
  - **VariantCaller -** Subfolders per sample ID containing the unfiltered genome VCF file; VariantCaller output logs
- **Results**
  - MergedMetrics
  - **<Subject ID>_DNA**
    - *coverage.thresholds.bed.gz*
    - *targets.quantized.bed*
    - *{SampleID} _DNA_CombinedVariantOutput.tsv*:
      The combined variant output file contains the variants and biomarkers in a single file that is based on a paired sample (if using PairID). The output contains the following variant types and biomarkers:
      - ❖ Small variants (including EGFR complex variants)
      - ❖ TMB

      The combined variant output file also contain Analysis Details and Sequencing Run Details sections.
    - *{SampleID}_DNA_CopyNumberVariants.vcf*
    - *{SampleID}_DNA_MergedSmallVariants.genome.vcf.gz*:
      The merged variant genome variant call file combines the small variant genome VCF (output of variant filtering) and clinically relevant variants in EGFR exon 19 from Phased Variant calling. This contains information on all candidate small variants evaluated.
    - *{SampleID}_DNA_TMB_Trace.tsv*:
      The TMB trace file provides comprehensive information on how the TMB value is calculated for a given sample. All passing small variants from the small variant filtering step are included in this file
  - {run_folder_id}_run_summary.xls
  - dsdm.json
  - MetricsOutput.tsv:

    The MetricsOutput.tsv file contains the following quality control metrics for all samples:
    - ❖ QC metrics for small variant calling (SVC)
    - ❖ TMB
    - ❖ RunQc analysis status and contamination

    This TSV file also includes expanded DNA library QC metrics per sample, based on total reads, collapsed reads, chimeric reads, and on-target reads. Analysis using RNA samples also produces RNA library QC metrics and expanded RNA library QC metrics per sample based on total reads and coverage.
- inputs.json

Figure 8. Examples of WES Output Files

CONCLUSION

As we continue to navigate the ever-expanding sequencing environment and Sponsors continue to utilize the benefits of customized output data, there will be growing requests to package reports in a manner conducive to FDA submissions. Third-party vendors may also face certain limitations with their data imports and will look to SAS programmers for their assistance in exporting the data in a way that the third-party vendors can then import with ease to quickly provide sponsors with the information they require to create a life-saving medication.

REFERENCES

SAS Help Center. (2024, January 31). *Macro Language Reference*. SAS Macro Examples. Retrieved April 1, 2022, from https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/mcrolref/n0ctmldxf23ixtn1kqsoh5bsgmg8.htm

Food and Drug Administration [FDA]. (July 2019).

Submitting Next Generation Sequencing Data to the Division of Antiviral Products Guidance for Industry Technical Specifications Document (FDA Docket No. FDA-2017-D-6821). Food and Drug Administration, FDA Guidance Documents. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/submitting-next-generation-sequencing-data-division-antiviral-products-guidance-industry-technical

ACKNOWLEDGMENTS

We would like to thank our colleagues in the Data Management department for their insights and guidance. We would also like to extend our thanks to our Eurofins Viracor BioPharma colleagues that took the time to review our paper and make this presentation possible. We would also like to thank PharmaSUG for accepting our paper and organizing this conference.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Oliver Lu
Eurofins Viracor BioPharma Services, Inc.
Oliver.Lu@vbp.eurofinsus.com

Katie Watson
Eurofins Viracor BioPharma Services, Inc
Katie.Watson@vbp.eurofinsus.com