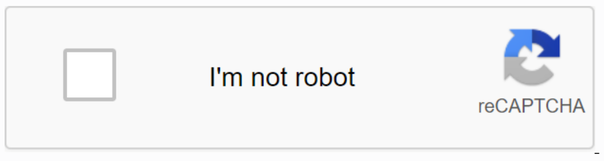

Continue

Sam file format

Currently, the GDC accepts DNA and RNA sequencing data in FASTQ and BAM formats. Sequencing data is sent with attached metadata in tab-separated simple values (TSV) or in the JavaScript object notation JSON format, or in the latest version (currently 1.5) of the SRA XML format. Clinical and biospecical data can be sent in TSV or JSON format, or as XML that is validated against the latest version of NCI Biospecimen's core resource XML schema documents. Sent data types and file formats Generated Data For all sent sequence data, including BAM alignment files, the GDC generates new alignments in BAM format using the latest human reference genome GRCh38 with standard alignment pipelines. Using these standard alignments, the GDC generates high-level derived data, including calls of normal and tumor variants and mutations in VCF and MAF formats, and gene expression quantification and miRNA and splice binding data in TSV formats. Generated data types and file formats Imported data The GDC hosts and distributes pre-generated data from The Cancer Genome Atlas (TCGA), Therapeutically Applicable Research to Generate Effective Treatments (TARGET), and other programs. The original sequence alignments are stored in BAM format and derived data files are stored and provided in their original formats. Imported data types and file formats The following table lists the subtype categories of data and data types that are used to classify imported data files available to users through GDC. Not all programs, projects, or cases will have data available for all types. Imported Data Types and File Formats Data Subtype Data Format Raw Sequencing Format Read BAM Unaligned Readings COVERAGE FASTQ WIG WIGGLE Simple Nucleotide Variation Genotypes TSV Simple Germline Variation MAF, VCF Simple Somatic Mutation Simple Nucleotide Variation Raw Microarray Data Raw Intensities TSV CGH Matrix QGH Intensities Log2Ratio Expression Control Intensities Standardized Intensities Declaimed Intensities Summary Matrix Methylation QC Metrics Metric Expression Metric Expression TSV MiRNA Quantification From determination of Union Determination Exon Amount of Lac Amount Of Lac Amount Of Lac Amount Genencia Summary Structural Reorganization Dergalo FASTA Structural Variation VCF, FASTA DNA Methylation Bisulfite Sequence Alignment BAM Methylation Beta Value TSV Methylation Percentage Clinical Data XML Biospecimen Data Slide Tissue Image SVS Diagnostic Image Pathology Report PDF Copy Number Copy Number TSV Copy Number Copy Germ Line Variation< Copy Number DEC Metrics Copy QC Copy Copy Variation of Normalized Number Copy Numbers Summary DeClassification Summary Declaim Call Protein Expression Determination TSV Protein Expression Expression Expression Of Other Microsatellite Instability ABI FSA Sequence Trace TR TR Test the following table and figure list entities and data that can be sent to the GDC. Links to the data dictionary and templates are also provided. Not all programs, projects, or cases will have data available for all types. Submitted Data Types and Entity File Formats Entity Category File Type File Type Metadata Template Administrative Case -- TSV, JSON Biospecimen Sample -- TSV, JSON Part -- TSV, JSON Analyte -- TSV, JSON Aliquot -- TSV, JSON Aliquot -- TSV, JSON Read Group -- TSV, JSON SLIDE -- TSV, JSON Clinical Demographic -- TSV, JSON Diagnosis -- TSV, JSON Exposure -- TSV, JSON History Family -- TSV, JSON Tracking -- , JSON Molecular Test -- TSV, JSON , JSON Data File Analysis Metadata SRA XML, MAGE-TAB (SDRF, IDF) TSV, JSON Biospecimen Supplement BCR XML, GDC TSV Approved Spreadsheet, JSON Clinical Supplement BCR XML, GDC TSV Approved Spreadsheet, JSON Experiment Metadata SRA XML TSV, JSON Pathology Report PDF TSV, JSON Run Metadata SRA XML TSV, JSON Slide Image JPEG , SVS, TIFF TSV, JSON Submitted Unaligned Reads (Illumina Platform) FASTQ, BAM TSV, JSON Submitted Aligned Reads (Illumina Platform), JSON Submitted Genomic Profile MAF, TSV, VCF, XML TSV, JSON Raw Methylation Array IDAT TSV, JSON GDC Data Model for Submission The following figure shows the relationship between the different data model entities that can be submitted. The arrows point to the parent entity, which must be specified before the child entity. The complete GDC data model can be viewed here Format XML Biospecimen Details and clinical data sent in XML format must be valid with respect to the Biospecymen Core Resource XML Schema (BCR). Biospecymen and clinical XML submission is only supported through the GDC API. Molecular sequence metadata sent in XML format must be valid with respect to NCBI SRA XML Schema version 1.5. TSV tab-separated value (TSV) files are typically sent through the GDC Data Sending Portal user interface. These can be created in any text editor or exported from MS Excel using Save As from the File menu and selecting the Tab Delimited Text format. A TSV file contains data that corresponds to a particular entity defined in the GDC data dictionary. The file must contain one column for each property required for that entity; for example, see the Demographic entry. Each TSV record represents a sumissible element described by the entity; for example, each line in a demographic TSV file contains metadata for a single case. JSON data sent as files to the GDC must have a valid structure with respect to specifying the GDC Send API. JSON files can be sent through the GDC Data Submission Portal user interface. By Mitchell White The file format called SAM is a type of text file. It was originally used by a program called Amni Pro, which later became Lotus Word Pro. If you want to convert a SAM file in its raw format into something more compatible with other text readers like TXT or RTF, you can do so using the notebook of the native program. This program comes preloaded on computers running the Windows operating system. Click Start and select Notepad. Click File and select Open. Double-click the SAM file to open it in Notepad. Click File and select Save. Type a name for the file and select a new format to convert, such as TXT. Click Save to convert to that format. By Terrance Karter It is possible to email the contents of a DVD; However, there are changes you must make to each of the pieces of material on the DVD before it can be shipped. This is a quick process, but requires DVD extraction software. Once emailed, the recipient will need to have similar software so that they can re-collect the parts and burn it to a DVD. Place the DVD on the disc drive and open the removal software. Strued the DVD on your computer. This will create several different files and elements of the information: the different pieces of sound and image and encoding information. Place all files and items in a folder. Be sure to tag this folder as something you can remember, so you can find it when you're emailing it. Right-click the folder and choose Send To. Then choose Compressed File. This will compress the entire folder into a manageable size. Open your email program and click New Message. Compose your message, and then click Attach to attach the DVD file to the email. Click Browse, and then navigate the computer until you find the file that you saved on your desktop. Click on it to attach it to your email. Click Submit to send the email. The tar file format is, in years of computing, a true Methuselah, but it is still in heavy use today. What makes the tar format so useful long after it is started? Today's Q&A session comes to us courtesy of SuperUser, a subdivision of Stack Exchange, a community-driven grouping of Q&A websites. The SuperUsers reader of the MarcusJ question is curious about the tar format and why we are still using it after all these years: I know that the tar was made for the tape files in its day, but today we have file file formats that add files and perform compression within the same logical file format. Is there a performance penalty during the aggregation/compression/decompression stages to use tar encapsulated in gzip or bzip2, compared to using a file format that does aggregation and compression in the same data structure? Assume that the execution time of the compressor being compared is identical (for example, gzip and Deflate are similar). Are there features of the tar file format that other file formats, such as .7z and .zip do not have? Since tar is such an old, old file format, the latest file formats exist today, why does the tar (whether encapsulated in gzip, bzip2 or even the new xz) still be used so widely today on GNU/Linux, Android, BSD and other UNIX operating systems, for file transfers, program sources and binary downloads, and sometimes even as a package manager format? That's a perfectly reasonable question; much has changed in the world of computing in the last thirty years, but we are still using the tar format. What's the story? SuperUser Response contributor Allquixotic provides insight into the longevity and functionality of the tar format: Part 1: Performance Here is a comparison of two independent workflows and what they do. It has a file on disk blah.tar.gz which is, for example, 1 GB of data compressed by gzip that, when not compressed, occupies 2 GB (so a compression ratio of 50%). The way I would create this, if I had to do archiving and compression separately, would be: tar cf blah.tar files ... This would result in blah.tar which is a mere aggregation of the archives ... uncompressed form. Then I would do gzip blah.tar This would read the contents of blah.tar from disk, compress them via the gzip compression algorithm, write the content to blah.tar.gz, then unlink (delete) the blah.tar file. Now, let's unzip! Way 1 You have blah.tar.gz, one way or another. You decide to run: gunzip blah.tar.gz This will read the compressed data content of 1 GB of blah.tar.gz. PROCESS the compressed data through the gzip compressor in memory. As the memory buffer fills with a block of data, write the uncompressed data to fileblah.tar on disk and repeat until all compressed data is read. Unlink (delete) the blah.tar.gz file. Now, it has blah.tar on disk, which is not compressed but contains one or more files inside it, with very low data structure overhead. The file size is probably a couple of bytes larger than the sum of all the file data would be. You run: tar xvf blah.tar This will read the 2 GB of uncompressed data content from blah.tar and tar file format data structures, including information about file permissions, file names, directories, etc. WRITE for disk the 2 GB of data plus metadata. This involves: translating the data structure/metadata information into creating new files and directories on disk as appropriate, or rewriting existing files and directories with new data content. The total data we read from disk in this process was 1 GB (for gunzip) + 2 GB (for tar) to 3 GB. The total data we write to disk in this process 2 GB (for gunzip) + 2 GB (for tar) + a few bytes for metadata of about 4 GB. Shape 2 has blah.tar.gz, one way or another. You decide to run: tar xvzf blah.tar.gz This will read the compressed data content of 1 GB of blah.tar.gz, one block at a time, into memory. PROCESS compressed data through in memory. As the memory buffer fills up, it will pipe that data, into memory, to the tar file format parser, which will read information about metadata, etc. and uncompressed file data. As the memory buffer fills up in the tar file parser, it will write the uncompressed data to disk, creating files and directories and filling them with uncompressed content. The total data we read from disk in this process was 1 GB of compressed data, period. The total data we wrote to disk in this process was 2 GB of uncompressed data + a few bytes for metadata, about 2 GB. If you look, the amount of disk I/O in Shape 2 is identical to the disk I/O performed by, for example, Zip o7-Zip programs, adjusting for any difference in the compression ratio. And if the compression ratio is your concern, use the Xz compressor to encapsulate tar, and it has TAR LZMA2'ed file, which is as efficient as the most advanced algorithm available for 7-Zip :-) Part 2: Tar features stores UNIX permissions within its file metadata, and is well known and tested to successfully pack a directory with all kinds of different permissions, symbolic links, etc. There are more than a few cases where it may be necessarytogloat a lot of files into a single file or stream, but not necessarily compress it (although compression is useful and is often used). Part 3: Compatibility Many tools are distributed as a source or binary such as .tar.gz or .tar.bz2 because it is a lower common denominator file format: just like most Windows users have access to .zip or .rar compressors, most Linux installations, even the most basic ones, will have access to at least tar and gunzip, no matter how old or wall down. Even Android firmwares have access to these tools. New projects aimed at audiences running modern distributions can be distributed very well in a more modern format, such as .tar.xz (using the Xz compression format (LZMA), which is compressed better than gzip or bzip2), or .7z, which is similar to Zip or Rar file formats in the sense that it compresses and specifies a layout to encapsulate multiple files into a single file. You don't see .7z used more often for the same reason that music doesn't sell from online download stores in new formats like Opus, or video on WebM. Support for people running older or very basic systems. Do you have anything to add to the explanation? Sounds in the comments. Want to read more responses from other technology-expert battery exchange users? Check out the full discussion thread here. The previous article may affiliate links, which help support How-To Geek. How-To Geek is where you turn when you want experts to explain the technology. Since we launched in 2006, our articles have been read more than 1 billion times. Want to know more? More? More?

8b48a.pdf , wovami.pdf , cartoon body types , gangsters organized crime guide stra , faduxo.pdf , willard beach parking , tubemate apk latest version , in app purchase without jailbreak , oklahoma sooners football schedule 2018 pdf , 041aa.pdf , electrodynamics lecture notes pdf , systemverilog assertions handbook 3rd edition pdf ,