

Using WGS Tools on FamilyTreeDNA BigY files

and other similar single-sequence BAM files

There is nothing special about the [FamilyTreeDNA BigY](#) BAM or VCF files. They are the output from typical bioinformatic tool chains and laboratory equipment used for clinical sequencing. Just pruned to only include the Y chromosome results found after alignment.

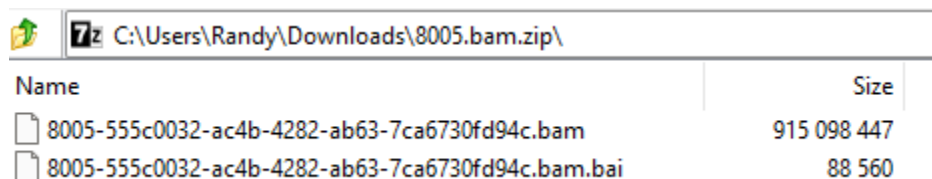
For the most part, everything mentioned here applies to Mitochondrial-only BAM files as well as Y and mitochondrial mixed BAM files. If you have a WGS or WES test result, consult traditional documents that support the whole genome reference models. But those much larger files can also be used in this way; especially if wanting to look only at the Y or mitochondria results.

We will start this overview with online, browser based tools and move finally to desktop tools. The initial focus is on the BAM file. But we will add the VCF file viewing and analysis later on as additional sections. For a deeper understanding of these and other tools and file formats, see the [additional documents](#) created to help you.

FTDNAs BigY product started life on their Illumina HiSeq sequencer. Then shortly before being relaunched as BigY-700, moved to the Illumina Novaseq 6000 sequencer. Both are commonly used in clinical testing laboratories to perform Whole genome and Whole exome sequencing (WGS / WES).

Prepping FTDNA y-Only BAM files

FTDNA delivers their BAM result in a ZIP file given a 4 digit code name. Think compression and folder / directory structure combined. Inside that ZIP file are two files with even more cryptic 32 character hexadecimal name following that code name. One file ending in an extension ".bam". The other in a ".bam.bai". The former file has your aligned sequencing results. The latter file is the BAM index file. You must extract these files so they are available separately on whatever device you are operating a browser on – be it phone, tablet or desktop computer. Often you can extract just by double clicking on the ZIP archive file. It is often useful to assign useful names as well. Just make sure to use the identical name before the extensions.



Name	Size
8005-555c0032-ac4b-4282-ab63-7ca6730fd94c.bam	915 098 447
8005-555c0032-ac4b-4282-ab63-7ca6730fd94c.bam.bai	88 560

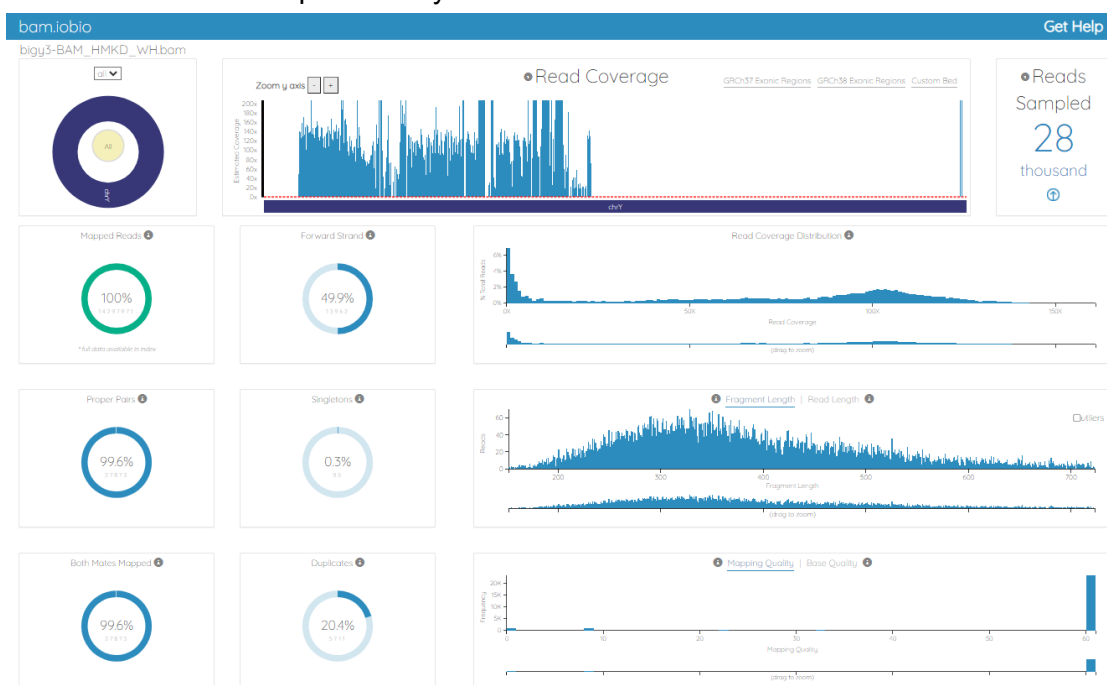
Note that since the introduction of the BigY-700 product, the BAM files are an add-on cost item. You must pay a fee of \$100 to get access to the file. A few days after ordering, they will provide a link to the file. You can either use that online link (good for only a week) in the tools below. Or download the ZIP file. It will be around 1 Gigabyte in size.

Using FTDNA BigY BAM files

So let's dive into various tools that can use your just downloaded BAM files right away.

BAM.iobio.io (University of Utah Medical Center)

You can get instant gratification with interesting stats and graphics using the site <https://bam.iobio.io/>. This site runs Webassembly versions of samtools and similar bioinformatics tools inside your browser. Doing random sampling to greatly reduce the overhead. Select "Local File" and then make sure to select both the extracted .bam and .bam.bai files. All data is kept local to your device and browser.



That big gap in the Y is the q12 region in the long arm that is nulled out of the Build 38 model. Over 30 million base pairs that are greater than 50% of the Y chromosome.

Fragment Length (insert size) and Read Length and most of the other stats are spot on. Notice the high duplicates rate -- this is due to that enrichment common in Exome (WES) sequencing; a technique similarly applied by FTDNA to target certain areas of the Y chromosome. The

duplicates are actually higher than indicated; these are only the duplicates already marked in the BAM file.

The Read Coverage distribution looks odd because it is an enriched sample like done for a WES test (see the information button next to the read coverage label). In the upper right, you can select a "Custom BED" file. See [BigY3 hg38.bed](#) made from looking at the 1 million SNPs delivered in the BigY-700. For the most part, all the non-N and non-PAR regions of Y. The BED file for BigY files do not really seem to help the read coverage with this tool though. We will work with the developer in Utah to investigate why. The Fragment Length is more inline with what we would expect after applying this BED file though.

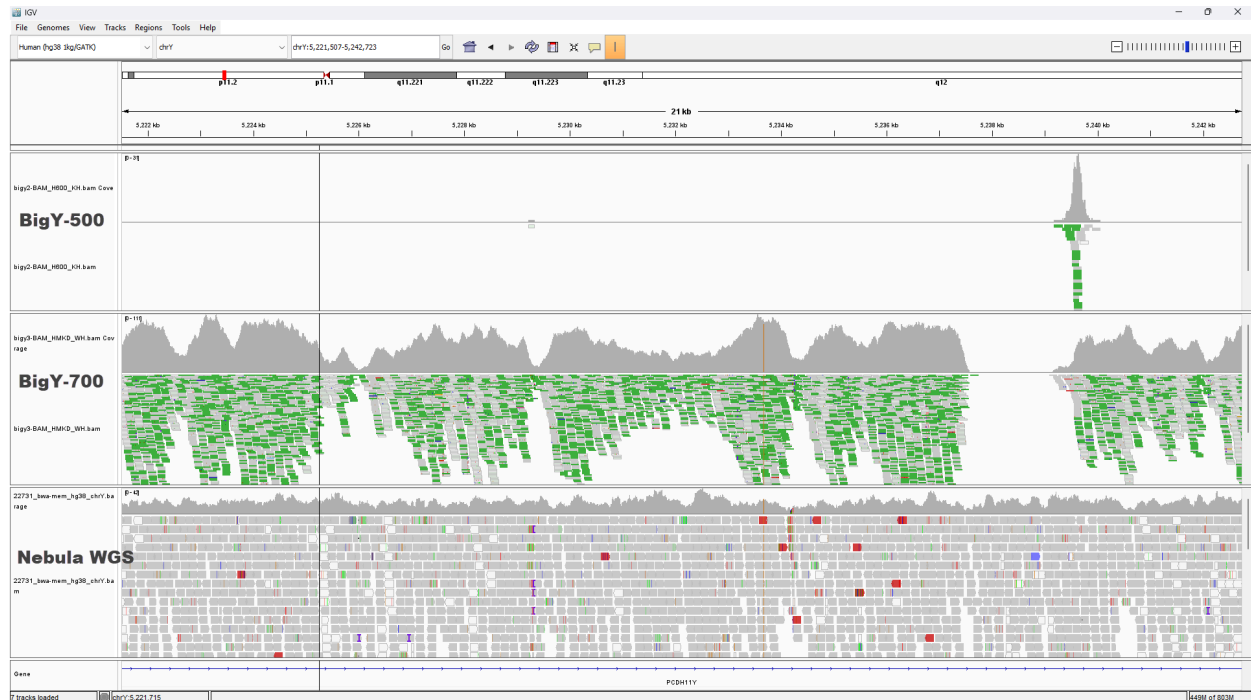
Genome Browser (that is, IGV)

Some like to visualize their BAM file in a Genome Browser. See the IGVWeb app at <https://igv.org/app/> for one such way. This is the same technology (likely same software) as FTDNA is using for their **Y-Chromosome Browsing tool**. Except now you are not restricted to just SNPs they named or the zoom level they depict. Genome Browsers are rich function tools that take in many formats and forms of information and try to display them in logical ways. But it takes a bit of effort to get the result in a form useful for each user.

First load the hg38 reference genome (the default is hg19 for this tool). Then select only the "ChrY" portion. Finally, zoom all the way in until you see only 40-70 base pairs at a time. Do this by click-hold and dragging over part of the ruler at the top or simply hit the plus (+) sign in the upper right 10 or more times or click the zoom gauge at the right most before the plus sign. Make sure you are seeing numbers in base-pairs (bp) and not kilo-base-pairs (kbp). The tool starts up showing the whole genome or whole chromosome which, for Y, is 59 million base pairs. Once zoomed in far enough, the reference model should appear in the upper track with colored boxes and/or letters.

Next load your BAM file (and its Index) as a Track. Make sure to select the `.bam` and `.bam.bai` (both) again. The gray "hilly" area at the top of this new track is the read depth per base pair. The layered bars below are the sequences (gray for forward, green for reverse; by default) stacked up as tight as can be visualized. You can load more than one BAM file, each into its own track, to compare them (visually).

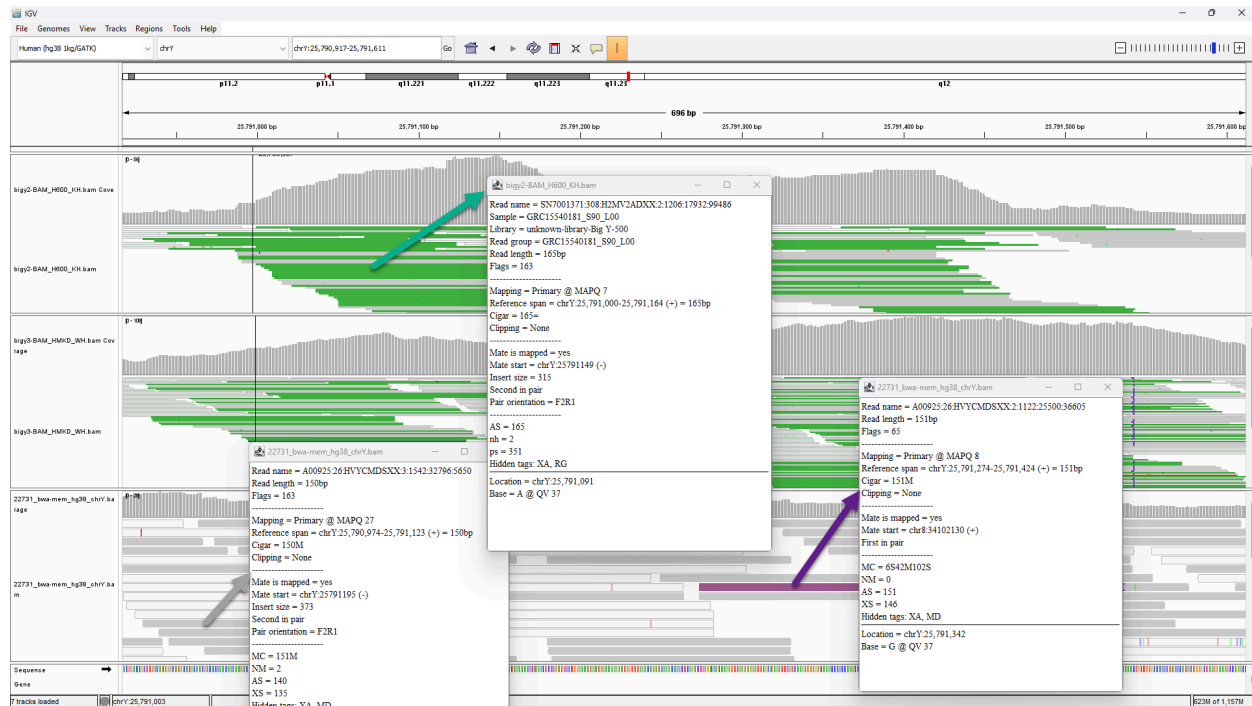
Most will want to search by known SNPs found in the Y. By default, this tool does not have that annotation available (as it is not common outside the phylogenetic tree / population genetics crowd to have these SNP names). We will update this document with where to get the ybrowse style annotation track to load. Likely it will require loading a new HG38 genome file with the annotation track added in as the web-based IGV does not allow custom annotation tracks.



Depicted above are three tracks loaded. A FTDNA BigY-500 kit, a FTDNA BigY-700 kit and a Nebula WGS. We have zoomed in to about 21 kb per screenful. This is typical as the -500 has less than 50% useful coverage, -700 closer to 75% and the WGS 99.9+% coverage.

You can click anywhere on the full Y chromosome at the top to jump to other areas of the chromosome. If you look at (zoomed in) the large q12 region in the long arm, you will see it is all nulled out with N's. And so no test kit will show any results in that area. Over 30 million base pairs long.

So for now, to search for a given SNP, you will have to reset the tool to display the region around the SNP of interest. Basically, you need to know the base-pair address for the SNP and enter the address +/- 50 base pairs in the track window select area at the top..



Above is a zoomed-in look where the reference sequence has just become visible at the bottom. We have clicked on three different read segments to get the details. Each represents one line in the BAM file. IGV default coloring happens to choose green for the second mate read on BigY files and white in the WGS file. Both show the first read in the pair as gray.

Note that deletions appear as fixed length, aligned “gaps” in the stacked read segments. Often with the read segments overlapping the gap. Insertions have a letter “I” appear in the track (often in bold dark blue). You have to click on the “I” to see the value inserted that does not appear in the reference. STRs will appear as insertions or deletions from whatever the reference default count is, if properly aligned and read. More complex variations cannot be represented.

This tool is running in your browser using javascript. All data stays local. But, as with the above tools, you need javascript enabled in your browser for this to work. You can easily download and run the IGV on your desktop. It does require a Java VM (JRE).

Other desktop or online browsers you can consider using are [JBrowse2](#) and the original UCSC Genome Browser (GB in a Box or [GBiB](#) for the desktop version). You can get helpful files to use with the browsers from the [yBrowse data directory](#). yBrowse itself uses an earlier version of JBrowse2 called GBrowse. Nebula and Dante have IGV built into their sites.

BAM.bio (that is, sandbox.bio)

Normally this would have been the second and next easiest step. But recent changes have removed the direct use and capability. But we still include it here for when the capability is added back.

As a next step, see the site <https://bam.bio/> which historically would allow you to run samtools in your browser but now with your own options. This site has morphed into a tutorial site <https://sandbox.bio/> with access to more than just samtools. ~~But it seems to have removed the ability to use a local file. We will update this document when we work with the developer to figure out how to still use your local file.~~ Likely you want to use the <https://sandbox.bio/playground> to get to the general tool. Beside a local file specification for input, you can now “download” the output to a local file. With the addition of tools such as htsfile, tabix and bgzip, you can now truly manipulate your files in your browser.

Samtools commands that can be nice to run are:

```
samtools view -H your.bam # to see the header
```

```
samtools view your.bam | head -3 # to see the first 3 lines
```

Note that the `samtools view` command is taking the internal binary BAM format and presenting it in the text SAM format.

WGS Extract (a desktop tool)

A next logical step would be to download **WGS Extract** (<https://wgsextract.github.io/>) to a desktop computer for use. This will not run in your browser as it is installing the full function, full capability bioinformatic tools for use locally. You cannot use this tool on a phone or tablet like the previous browser-based tools.

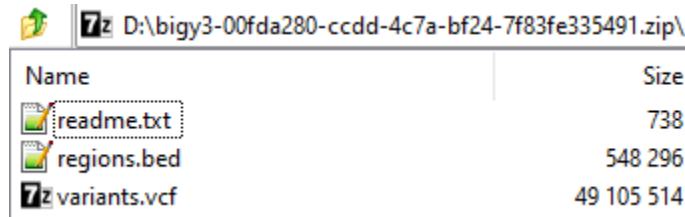
With yDNA-only BAM files, you can get the extended statistics, the yLeaf Haplogroup analysis (using the ISOGG tree), and even generate an Annotated VCF. The annotated VCF includes the SNP names so you can search using a text editor or similar. Included in the statistics is a bin coverage analysis to get a real sense of the coverage over the readable Y chromosome. When you select “WES” coverage, it uses the CombBED / McDonald / Poznik regions BED file to provide you coverage over just the regions used for age-determining SNPs.

Once you get familiar with this tool, you will be ready for its next update which will allow you to remap your own BigY BAM file into the new T2T model and start your new analysis there 😊. In the meantime, the tool does install all the bioinformatic tools for you. Simply drop into a Terminal / Command shell to run them directly.

Try the documents listed at <https://h600.org/wiki/Bioinformatics+Documents> for additional information and further possible steps.

Using FTDNA BigY VCF files

The VCF file is also delivered in a ZIP with the actual .vcf file taking the bulk of the space inside. An extracted BED file and readme text file is also added in.



Name	Size
readme.txt	738
regions.bed	548 296
variants.vcf	49 105 514

The regions.BED file is defining where FTDNA has extracted your VCF entries from. A replacement for the BED file mentioned earlier, if you like. The key file is the variants.vcf. Again, extract the files individually and rename as needed. Your BAM of before was near 1 gigabyte in size. This VCF is much smaller at 50 megabytes or smaller. If for some reason it is not compressed, make sure to compress it before supplying it to other tools. It must be in gzip format (or optimally bgzip). 7zip on Windows is a useful utility to get the Unix gzip format. It is native on Linux and MacOS.

Genome Browser on VCF Files

Yes, the IGV browser can be used on VCF files as well. Just that instead of viewing the layered read segments, as in a BAM, you are seeing point SNPs and any other variations contained in the file.

Targeted Y Phylogeny Tools

While not really WGS tools but targeted Y phylogeny tools, we mention these additional tools for use with BigY VCF and BAM files.

Cladefinder by Hunter Provyn (on the ySeq site)

Want to see where you would be placed on the yFull tree? Even before ordering your BAM file for upload there? Go to <https://Cladefinder.yseq.net/> and scroll to the bottom where you can specify your VCF file to upload. Within a few minutes you will get your clade (haplogroup) and a number of useful links to other tools and sites (Hunter's Phyloagrapper mapping program, the yFull site itself, etc).

yFull Phylogenetic Tree of Haplogroups

You can (initially) upload your VCF to <https://yFull.com/>. This will get your sample placed on their tree. But much of their analysis cannot be performed until you provide the BAM file. For example, STRs cannot be determined from the SNP-only VCF file supplied by FTDNA. Make sure to separately upload your y111 STR results.

Many using WGS test results will create a Y and Mitochondrial BAM and upload that. Then they are placed in the Y and mitochondrial trees.

ydna-warehouse

(formerly

Appendix: Getting jBrowse2 and IGV working for SNP name search with yBrowse data

Simply a place to catalog what it is taking to get the two popular desktop genome browsers to work with Y-only BAM files with SNP and STR name search. Also applies to the online versions of the browsers. The two being checked are IGV and JBrowse2. Also existing are GBiB, JBrowse, and GBrowse. Online is the original UCSC GB.

For IGV, you have to create a genome definition file to load (a) the hg38ChrY-only FA file, (b) a cytoband file (for visual clues), and (c) GFF3 files for STR and SNP names for search. The needed files appear to already be in <https://ybrowse.org/gbrowse2/gff/>. Which would imply simply creating a JSON file that could reside there (or reference those remote URL-based files) to enable the needed features. Note that you still have to load your own alignment (.bam and .bam.bai) file as a separate track.

For IGV, see the [Reference Genome Format](#), [JSON Genome Format](#) and [online AWS genome JSON file](#). Except for the FASTA file and its index, the rest are likely to be in GFF3 format. See the [IGV GFF3 summary](#) and [GFF3 specification](#). Eventually, most reference back to the original UCSC Genome Browser and its [defining file formats](#).

If wanting a solution that works for the web and desktop app, IGV must have the genome defined in the newer .json format and not the old .genome one. See the [IGV issue answer](#) about this.

Issues:

1. The `snps_hg38.gff3` file is over 500 MB in size. Too large for the tools here to load. Instead, they require large files be bgzip compressed and TABIX indexed. (Tabix and BGZIP being htlib utilities.) Being indexed, they must be coordinate sorted as well (which it appears this file is not).

The problem is that the `id=` and `name=` entries, used for search, do not appear in the index file. Only the index file is loaded and then used to step into the necessary area of the original, compressed gff3 file to selectively read the section needed. So any gff3 supplied as compressed cannot then have its `id=` and `name=` entries searched on.

The solution is to fix the original SNPS GFF3 by sorting it, changing characters (see below), compressing and then indexing it. Then, make a subset GFF3 without any other column 9 entries than `id=` and `name=` that is uncompressed like the original that can then be added as a hidden track to enable the name search. This same technique, with a separate hidden track per chromosome, will likely be needed in the eventual dbSNP enabling entry to be made for WGS files.

Ted Kandell of open-genomes.org has gone the route of using a PERL script to change from the gff3 to a BED format file with the smaller subset, only required information.

What is the largest gff3 file that can be loaded into a track without tripping the warning or complaining of not finding an index file?

2. **The str_hg38.gff3 file works as is** (small enough to not need compression so enables search by name functionality). But there is a functionality issue where the start and stop are based on the primers to find the STRs and not the STR itself. The [BAMSaway Chrome Extension](#) is better at defining the exact model coordinate position where the STR starts. Need to rework this file to have the STR coordinate start position (even if reverse strand defined). There are 402 STR entries in this file. BAMSaway has the y111 but not sure which others. yFull and FTDNA have minimal overlapping sets of over 700 STRs each. Where does one get the more complete information to create a more complete and useful file here?
3. **The cytobands_hg38.gff3 file is not the expected format for the JSON cytobandURL entry.** It wants a simpler, TSV text file that is zero based. Also, it seems the comments allowed in the GFF3 file are not allowed in this simple TXT format file. The yBrowse cytoband file explicitly defines the centromere region, which is useful. The hg36 model at the IGB AWS site spans this region. So edit the yBrowse cytoband to be zero-based, simple format, no leading / header comments. Rename to **cytobands_hg38.txt**
4. **GFF3 files have additional reserved characters in the column 9 list of name=value attributes that are used incorrectly in the snps_hg38.gff3 file.** Clearly, no tab, equals sign, nor semicolon can be used except as defined to separate columns and such. A few more key ones are no percent (%), no ampersand (&), no comma (,) except specifically where required in xxxx, and no trailing semicolon (;) (only to separate an attribute; not as a terminator). All these special characters, if used inside column 9 where not specified as required, must be [URL encoded](#). (As if the column 9 field were to be passed as a URL string from browser to server and back.) This is specifically defined in the [GFF3 spec](#).

You cannot simply double quote wrap values in name=value attributes. This does not escape the special characters nor is it allowed. No escape characters like backslash (\) used in shells or the plus sign (+, from initial GFF definition) are allowed either. The only solution, if a special character is needed, is to URL Encode it to the hex ASCII value. So the percent character becomes %25, the ampersand becomes %26, and so on.

The plus (+) was used freely in the comment named attribute of column 9. The ampersand (&) in the yfull_node= attribute and xxxx. yfull_node= was mostly (always?) the last attribute and sometimes had a trailing semicolon (;) which is not allowed. Not clear if an equals sign appears in the comment= value body and needs to be escaped. Some hints at using RegEx in a text or stream editor to clean this up are given in a [StackOverflow](#)

[answer](#). The simple solution, to do programmatically, is to do a URL.encode on the value string.

5. **What RefSeq file should be used for this work?** Is it even really needed. Normally used to define the genes in the chromosome. But we are not interested in genes for this work. But does IGV have additional, special properties it associates with this track?. Including when and how to display the reference sequence contained in the FASTA file itself.

More importantly, what format is expected given their example files have a .txt extension and do not follow other formats such as gff3? See an [IGV issue/answer](#) posted about how IGV uses the filename to figure out the file type; not the file extension like normally done. But the .txt file format used is not defined by the filename or extension. Seems the JSON track entry "format": "refgene" is used to define that the referenced file is in this special format. Not (or additionally) the file name.

Our solution here is to edit the [existing WGS hg38](#) file in the IGV base reference to just include the Y.

6. **Can the reference model (fastaURL) be bgzip compressed and indexed?** This would seem to make sense as it is the (near) largest file needed. But some testing is showing a compressed version, even with the index existing, is not working properly. Need to characterize the issue and figure out how to use it. Could be more a random-access into a large file in the browser protocol.
7. **Do we need a separate JSON file for each of the main Build 38 reference model classes (hg, 1k, ebi, etc)?** The answer is likely yes given the FASTAs are different between them.
8. **Should we add an N region annotation track** to make it clearer, when zoomed out, that the reference is all N's in a large area? What about BED track files for Poznik / CombBED / McDonald regions?
9. There are **parallel files for the mitochondria sequence** on yBrowse in the same directory. They should be similarly modified into compliance and tacked onto the ChrY ones already mentioned above. Then this single JSON file can serve to be used for Y-only, M-only and YM BAM files; like would typically be found from non-FTDNA BigY tests (that is, WGS processed for yFull or similar analysis).

Current draft JSON file for yBrowse SNP name search in Y only BAMs loaded into IGV

```
{
  "id": "hg38ChrY",
```

```
"name": "Human (GRCh38/hg38) ChrY Only",
"fastaURL": "hg38ChrY.fa.gz",
"indexURL": "hg38ChrY.fa.gz.fai",
"cytobandURL": "cytobands_hg38.gff3",
"aliasURL": "hg38ChrY_alias.tab",
"chromosomeOrder": [ "chrY" ],
"tracks": [
  {
    "name": "yBrowse SNPs",
    "type": "variant",
    "format": "vcf",
    "url": "snps_hg38.vcf.gz",
    "indexURL": "snps_hg38.vcf.gz.tbi"
  },
  {
    "name": "yBrowse SNPs",
    "type": "annotation",
    "format": "gff3",
    "url": "snps_hg38.gff3.gz",
    "indexURL": "snps_hg38.gff3.gz.tbi"
  },
  {
    "name": "yBrowse STRs",
    "type": "annotation",
    "format": "gff3",
    "url": "str_hg38.gff3"
  },
  {
    "name": "yBrowse Palindromes",
    "type": "annotation",
    "format": "gff3",
    "url": "palindromes_hg38.gff3"
  },
  {
    "name": "Refseq Genes",
    "format": "refgene",
    "url": "ncbiRefSeq_hg38ChrY.txt.gz",
    "indexURL": "ncbiRefSeq_hg38ChrY.txt.gz.tbi",
    "visibilityWindow": -1,
    "removable": false,
    "order": 1000000
  },
  {
    "name": "Search SNPs",
    "type": "annotation",
    "format": "gff3",
    "url": "snps_hg38_100k.gff3",
```

```
        "hidden": true
    }
  ]
}
```

If placed in the same directory as the referenced files, they will all be picked up together after loading. Needs to be tested and confirmed when a remote URL is used to load the original JSON file. (Some early testing indicated many of the file types were failing to be randomly indexed and loaded the way IGVWeb does with its AWS based files. Is there some special setting on the web server needed to enable this type of access?)