Making links to the UCSC Genome Browser. Part 1: Dissecting / Understanding the URL

Welcome to our tutorial describing how to make links to the UCSC Genome Browser. This is Part One of a three-part series exploring how to link to the Brower for a variety purposes.

[0:17] Why make URL links?

There are many reasons you might want to make links by direct URL. If you simply want to share a link with a colleague that shows some location with certain tracks turned on, you can quickly construct it directly in the URL. It's fairly simple and does not make a permanent record in your Saved Sessions.

You may wish to have some links in your bookmarks to key places in the genome. Because URLs tend to be additive, if you have data tracks turned on and click a link that simply points to some gene, the links we describe here can be designed go to the other location, leaving all other datasets in your display in the same configuration. Any datasets you add to the link will be added to whatever setup you already have.

You may be developing a web site which generates information about locations, genes, or other annotations. Your software can generate a link to the Browser on the fly that shows exactly what you wish to show. You define the location, which datasets are turned on and how they are configured.

You may have custom tracks or a track hub on a server somewhere and you wish to easily load the data.

Or you may have a spreadsheet with many rows of coordinates, gene names or reference SNP identifiers (rsIDs). You can easily make a link that you copy down an entire column, creating custom links for each spreadsheet row.

[1:41] Saved Sessions vs URL links

An excellent way to save and share stable Browser views that contain exactly what you wish to show is the Saved Sessions feature. Saved Sessions require a login and they create a semi-permanent link that we store in our database for future access. I say "semi"-permanent because, while we attempt to preserve your sessions and even back them up, you are encouraged to keep a local copy, just in case. Hardware happens. The process for managing Saved Sessions is discussed in a previous tutorial in the UCSC video collection, bit.ly/sessionVid.

Saved Sessions have the property that they completely overwrite any tracks you may have open and will set the Browser to exactly the setup as when they were saved. The links we will discuss in today's tutorial will typically add tracks to whatever you have open (unless you use the hideTracks parameter discussed later).

[2:40] About the URL

Today we will focus on making URLs on the fly. These URLs are distinct from Saved Sessions in several ways, but typically, they are additive, allowing whatever tracks are already on in the Browse to remain on, simply adding to them as defined in the URL. We'll also show you a parameter that defeats this tendency (hideTracks).

[3:02] The hgsid cookie

We will start by going to the Browser at the default location. “Reset All User Settings” in the top bluebar sets us at the hg38 human genome assembly. We will use hg38, but most of what we will see is applicable to any genome assembly. The [go] button will take us into the graphical viewer.

One of the first things you will want to know is, you cannot simply copy an existing URL and send it to someone. The URL contains a string of characters that refers to a cookie in the UCSC system. That’s how the Browser knows how to get back to your location when you come back. If you close the Browser and come back in a few days, we look up your settings based on the cookie saved on your machine and return to your *last* position and settings.

If you send someone a link with the hgsid (session identifier) cookie in it, the link will continue to follow you around, resetting every time you do anything on the Browser. When a link with an hgsid is clicked at a later time, your most recent view in the Browser will show up, *not* the place you think you saved and shared.

Let’s demonstrate that. First, let’s simplify the screen, by hiding all the data tracks, then turn on the GENCODE Genes track. We have the ACE2 receptor gene in the window. The product of this gene happens to be bound by the coronavirus spike protein during infection by SARS-CoV-2 virus. And we’ll turn on the SNPs track and set it to pack.

Watch this: I’ll take the URL and copy it to a text page to preserve it. Perhaps I want to share this view with someone and I emailed her this link. I want her to see exactly this view: The genes track and these SNPs on the ACE2 gene.

Then I get busy looking at other genes, IL2, maybe and other data, and I turn on the CRISPR Guides and Conservation tracks and the new ENCODE cis-regulatory elements track. I turn off the SNPs. My colleague clicks the link I sent her. I’ll switch from Firefox to Chrome to simulate another person. We both expect that she’ll see the ACE2 receptor and the SNPs in the window. No such luck. She’s looking at ACE2, all right, because the position is encoded in the URL, but the tracks are all wrong. She’ll see the tracks that match the cookie – which records what I did *after* saving and sending the link, *not* what I wanted to share. The cookie is tracking what I’m doing. The SNPs are gone.

[5:52] Hacking the URL

Let’s dissect the URL we saved and use that information to build a URL that will do what we want.

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chrX%3A15560138%2D15602945&hgsid=847166315_zfBBKPvZgFzFExwFynP3z3D9

We’ll pull it apart into its key-value pairs. Each is separated from the others by an ampersand.

Starting from the end, we’ll dissect it in pieces. The session ID we’ll want to throw away, as we just discussed.

```
&hgsid=847166315_zfBBKPvZgFzFExwFynP3z3D9
```

The position parameter is next:

```
&position=chrX%3A15560138%2D15602945
```

This is simply an encoded form of <chrom> colon <chromStart> hyphen <chromEnd>. You can use the actual, more intuitive colon and hyphen when you make your own URL instead of the encoded versions. For example, what you see here:

```
&position=chrX:15560138-15602945
```

where

```
%3A is replaced by a colon
%2D is replaced by a hyphen
```

There are other ways to define position that we'll get into presently, but you can always define a location by genomic coordinates.

The next set of parameters in the URL:

```
&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&
nonVirtPosition=
```

all have " VIRT" in the name. These settings are not relevant to the current discussion. They relate to virtual chromosomes created by Multi-Region Mode, which is the topic of a three-part series of videos in the Browser collection.

Finally, the first parameter after the question mark:

```
db=hg38
```

defines the genome database we are using, human hg38.

The base URL:

http://genome.ucsc.edu/cgi-bin/hgTracks?

identifies the main Browser graphic. Everything discussed here also applies to making links to our genome-euro and genome-asia mirrors. Simply add the hyphen and locality name after the word "genome." The question mark at the end of the base URL defines the beginning of the key-value pairs. You can think of them as the variables and their settings.

Let's build a URL from the bottom up. We'll start by building a URL that turns off all tracks by adding

```
hideTracks=1
```

to the base URL and database name, giving us:

http://genome.ucsc.edu/cgi-bin/hgTracks?hideTracks=1&db=hg38

When we use this in a web browser, nothing is showing but the Base Position/coordinate track. All data tracks are off.

[8:30] Building a URL: Identifying track names

Let's add back the GENCODE Genes track and in the process learn how to identify the tablename for any track. One way to get the tablename is to mouse over the label in the track controls and look for the variable "g." You see that the tablename for the GENCODE Genes track (which is called UCSC Genes in hg19) is "knownGene."

We can set

&knownGene=pack

to load the track. With no "position=" parameter, the track opens at whatever position we were at before. Note that all of the parameters are case-sensitive. Table names are often several words strung together using interCapping, more commonly called camelCase, where everything is lower case except the first letter of internal words.

We can make the same URL for hg19. On hg19, the knownGene table opens the UCSC Genes track.

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg19&hideTracks=1&knownGene=pack

Here we have substituted hg19 in where hg38 was before. This particular, bare-bones link can be very useful if you make this a link on your browser toolbar.

[9:43] Toolbar link – Turn off all but genes track

Here I have made a toolbar link, labeled "genesOnly" that will turn off all tracks except the track specified by the table, knownGene. Leaving out the database designation will make the link work for any assembly you happen to be viewing if it has the table, knownGene.

http://genome.ucsc.edu/cgi-bin/hgTracks?hideTracks=1&knownGene=pack

For hg38, this will turn on GENCODE Genes at your current location; for hg19, it will be UCSC Genes. It will also work in mouse.

Watch as we move to the gene NF1, turn on all the default tracks using the button below the Browser graphic, then use the link. See? Gene track only. There is nothing to stop you from adding your favorite tracks to such a link so those tracks will come on as well when you click the link. We'll see in a moment more about to control the display of any track you wish to see.

[10:49] Adding in tracks of choice

Let's add the OMIM Alleles track from the Phenotype and Literature group, using tablename "omimAvSnp." We can set a track to any visibility level in the dropdown menu. We'll use "pack."

&omimAvSnp=pack

and return to hg38. This gives us the URL as shown here:

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&hideTracks
=1&knownGene=pack&omimAvSnp=pack

There we are: Two tracks are turned on (albeit there are no data in the OMIM Alleles track at the ACE2 gene). Anyone clicking such a link would have those tracks turned on at whatever location they were using last time they had the Browser open. To demonstrate, let's go to the Saved Session:

http://genome.ucsc.edu/s/videoDemo1/hg38_links1

You can see the Common SNPs and ClinVar tracks turned on. We're now at the EGFR gene, as was saved in that session.

Now let's reload the link that turns on the OMIM Alleles.

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&hideTracks
=1&knownGene=pack&omimAvSnp=pack

SNPs and ClinVar are now off because of the "hideTracks" parameter; only the genes and OMIM Alleles are on.

Let's redo it without the "hideTracks" parameter. Load the session again:

http://genome.ucsc.edu/s/videoDemo1/hg38_links1

and go to a different gene, park7. Now we'll load that earlier URL, removing the "hideTracks" parameter.

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&omimAvSnp=pack

The result is to add the OMIM Alleles track, but without removing the existing SNPs and ClinVar tracks.

[12:43] Links with gene names, not coordinates

Now let's add a parameter to jump to the gene by name, so you don't have to determine the coordinates. Another nice thing about this feature is, you can make two URLs that are identical except for the database parameter, "db," and can go to your gene in either genome assembly without having to know anything about the mapping of coordinates from one assembly to the other.

You can find the syntax for this command, as well as many other useful parameters and their values on the FAQ page for links:

http://genome.ucsc.edu/FAQ/FAQlink.html#gene

We will add

```
&singleSearch=knownCanonical&position=SOS1
```

to get us to the SOS1 gene.

Note that the knownCanonical table is where we store the "best" isoform for each individual gene locus. Using singleSearch on the knownGene table instead of knownCanonical would lead to an intervening disambiguation page, because knownGene has an entry for each isoform and you would need to choose which one to visit, whereas knownCanonical has a single entry per gene that leads directly to the Browser.

That results in this link which hides everything but GENCODE Genes and OMIM Alleles and goes to SOS1:

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&hideTracks
=1&knownGene=pack&omimAvSnp=pack
&singleSearch=knownCanonical&position=SOS1

[14:11] Other identifiers

Many identifiers can be used with the position parameter directly. If the format of an identifier is unique to a track, such as the OMIM Allele identifier, you can set the position to it directly, without the singleSearch parameter, such as

```
&position=131550.0007
```

If hideTracks is turned on, you should name the table because tracks will be turned off even though you arrive at the correct location.

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&knownGene=pack
&position=131550.0007

The mouseover shows you what this is: Glycine to Aspartate at amino acid 428, causing "inflammatory skin and bowel disease, neonatal."

Note that the searched-for item name is highlighted, and has 50 bases of padding on either side.

[15:06] Links with SNP rsIDs

Let's pick up an rsID from dbSnp and use that for navigation in a link. "rs" is "reference SNP," though these days it is more fashionable to call them SNVs rather than SNPs, as the word "Polymorphism" has implications (specifically, that minor allele frequency is above 1%) that do not apply to most of the variants in the database. But "sniv" is difficult to pronounce and many people use "snip" synonymously. The most recent release, snp153, is a composite track, a more complicated situation, which we will discuss in a later video in this series. We'll turn on the Common SNP 151 track, though in an exon of an important gene, we would not expect to find many Common SNPs.

As expected, we find no Common SNPs in this window, so let's zoom out by 100x. There is one SNP over here that is red, meaning that there is an amino acid change, so let's click into that one to copy the rsID. While we're here, you can see the change: We have a G to A transition, with A having a minor-allele frequency of 29%. The variant leads to an arginine to lysine change. That's at least a conservative change of one positively charged residue to another, perhaps explaining why it can be so common in the population.

Let's copy the rsID, rs2227983. The low rsID is consistent with the allele frequency – Common SNPs are more likely to have been found in the early days of sequencing and have lower numbers. Now we can add the SNP to a URL as the value for the position variable and go there.

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&knownGene=pack&position=rs2227983

We can also just add "&position= <rsID>" to the existing URL. Let's try that.

We'll have to choose which track to turn on, as this SNP appears in many tracks. The higher the dbSNP release number, the more recent the data release. The Common SNPs are a subset of the All SNPs dataset. Let's select the release 151 All SNPs track by choosing: "Simple Nucleotide Polymorphisms (dbSNP 151)"

You can see that the SNP we included in the URL is marked in reverse video with 250 bases of padding on either side. The SNP now appears in two tracks: Common SNPs, which was already turned on, and All SNPs, the one we clicked. And it is highlighted, though very faintly. The highlighting is easier to see when we zoom in.

You can get around the disambiguation page if you use singleSearch, which we saw earlier in the context of searching for genes. Specify the search table as snp151 and the position using an rsID, so we have:

&singleSearch=snp151&position= rs2227983

http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38
&knownGene=pack&singleSearch=snp151&position= rs2227983

Let's turn off the snp151 track and use the URL to confirm that it will turn back on. We see that it does. If you use hideTracks=1 in your URL, then you must also turn on the snp151 track explicitly if you are using singleSearch. Otherwise, you will go to the correct location, but the track will be hidden.

This concludes Part One of the three-part series of videos showing how to make links to the UCSC Genome Browser. Part Two will show how to navigate to specific locations within genes without knowing their genomic coordinates, how to set highlights and how to use some other useful features of making customized links to the Browser.

In Part Three, we will discuss how to load composite tracks, how to find obscure configuration parameters, how to access your remotely-hosted custom tracks and track hubs and how to make useful links in your spreadsheets.

Thanks for watching and thanks for using the UCSC Genome Browser. You can subscribe to the UCSC Browser YouTube channel and watch for new installments at

https://bit.ly/ucscVideos

This video was released during the global coronavirus pandemic. Please note that when it is safe to travel again, we will restart our on-location training program. You can contact us to host a one-day or two-day in-person training at your location at

https://bit.ly/ucscTraining