

## **DeepView Manual**

2014-04-10

Jiyuan An, John Lai, David Wood, Atul Sajjanhar, Chenwei Wang, Gregor Tevz, Melanie Lehman, Colleen Nelson: Australian Prostate Cancer Research Centre (APCRC-Q) and Institute of Health and Biomedical Innovation (IHBI), Queensland University of Technology (QUT), Brisbane, Australia  
[j.an@qut.edu.au](mailto:j.an@qut.edu.au)

Web page: <http://www.australianprostatecentre.org/research/software/DeepView>

## 1. Installation

### 1.1. Java Requirement

DeepView requires Java 6 or later be installed at your computer. From <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, you can download JDK(Java SE Development Kit) to install.

### 1.2. Command Line start

- a. DeepView related files are zipped at the URL:  
<http://www.australianprostatecentre.org/research/software/DeepView>.
- b. Download the zipped file into a directory – e.g. C:\DeepView (Windows), or /home/xxx/DeepView (Linux).
- c. Once the file is unzipped, a new directory “DeepView” will appear. The directory contains four files:
  - **DeepView.bat**, which is a batch file, which runs in a Windows environment. A shortcut can be created on the desktop to allow DeepView to run by simply clicking the shortcut.
  - **DeepView.sh** and
  - **DeepView.jar**, which stores java classes of DeepView. This is the main file to run this tool.

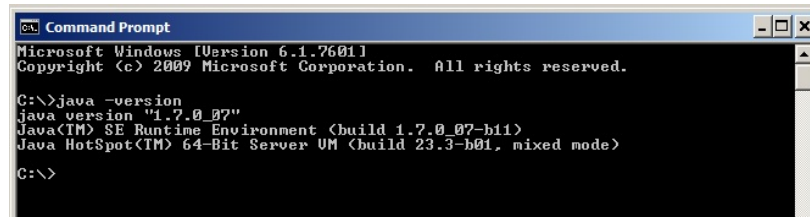
And two subdirectories:

- **Lib directory**, which contains all java libraries, used in DeepView, such as RNAfold and picard libraries.
- **Data**. Under the data directory you will find:
  - **init.xml**, which contains which genome will be visualized in the genome browser.
  - **Genome folders** (hg19, hg18, mm10). Each folder has the following files.
    - **Track.xml**, which contains information for the genome browser configuration and track. It is in XML (Extensible Markup Language) format. All items in the track.xml file, such as file name and colour, are enclosed within a pair of tags. In the beginning of XML file, the chromosome size of the target genome is listed. The tags, <locus> and </locus>, are used to set the chromosome coordinates in the genome browser. For example:

```
<locus>
  <chr>chr19</chr>
  <loci_start>51409009</loci_start>
  <loci_stop>51414310</loci_stop>
</locus>
```
    - **Track files** such as BED, wiggle, and BAM files,
    - **genomeData** which is zip file contains genome sequences. Genome sequences are fasta format and each chromosome corresponds to one fasta file.
    - **Reference files** which have “tx” extension. Their index files (xxx.idx) will be created for their first time use.
  - **Demo folder**, which contains demo track data files such as bam, bed and wiggle files.

### 1.3.1 Start DeepView

- a. First ensure that java is executable, as in Figure 1. If not, you need to check the system variable path.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\>java -version
java version "1.7.0_07"
Java(TM) SE Runtime Environment (build 1.7.0_07-b11)
Java HotSpot(TM) 64-Bit Server VM (build 23.3-b01, mixed mode)

C:\>
```

**Figure 1 test whether java is installed**

- b. In Windows, double click on the DeepView.bat (800MB memory) or DeepView\_2G.bat (initializes 2GB memory) file to start DeepView. You can also go to the command prompt to start DeepView manually.
- c. In Linux or MacOS, type: >java -jar -Xms2g DeepView.jar , or .sh file. For example >./DeepView\_2G.sh, which initializes 2G memory for DeepView.

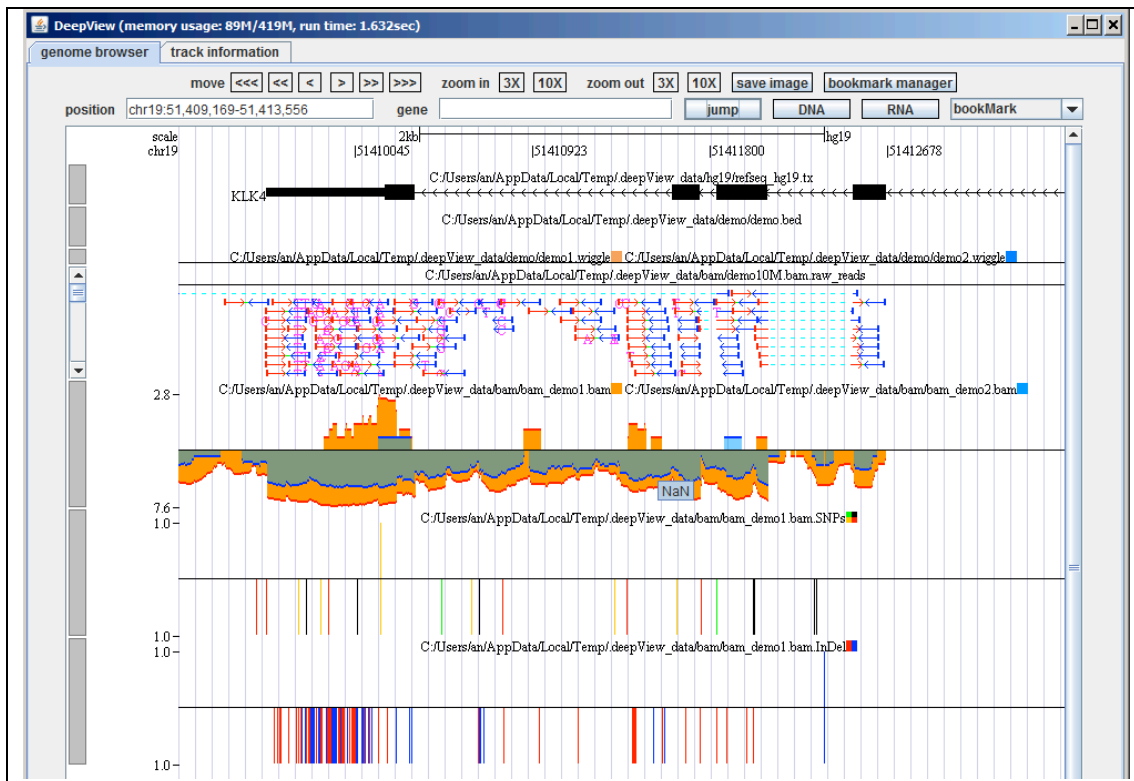
The option -Xms (initial memory allocation pool) is set according to your machine memory. More memory is needed if you load very deep RNAseq data.

DeepView interface has two tabs: one for the genome browser window, a second for the track information window.

### 1.3.2 Genome Browser

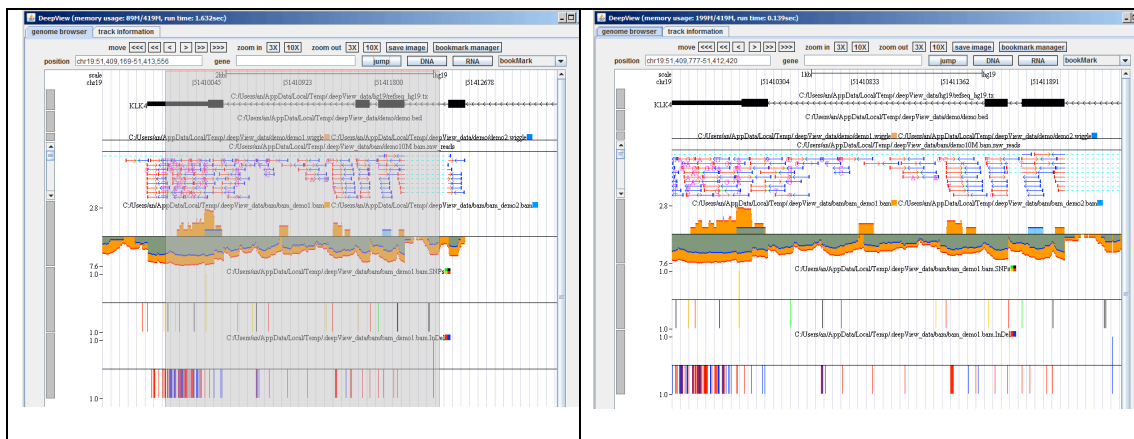
Once DeepView has started, six tracks (as indicated in track.xml) are shown in the genome browser (Figure 2), where most buttons are functionally similar to that of the UCSC genome browser.

- a. A track will be highlighted if the mouse pointer moves over on the left margin of the track.
- b. Moving across the genome is achieved by pressing the <<<, or <<, or <, or >, or >>, or >>> buttons as follows:
  - < and > move the browser left or right by 10% of the genomic region currently displayed.
  - << and >> move the browser by 50% of the displayed genome, and
  - <<< and >>> move the browser by 100% of the displayed genome.
- c. Zooming can be done with the 3X and 10X zoom in and out buttons, or by placing the mouse pointer at the top of the canvas of the genome browser, then clicking and holding the left mouse button. With the left mouse button held down, drag the pointer across to your locus of interest, and then release the left mouse button. This will enlarge the area of interest into the full canvas as shown in Figure 3. The highlighted region on the left is shown as the selected full-sized region on the right.
- d. Tracks can be rearranged in the genome browser window. To reorder tracks, click and hold the left side of the track and drag the highlighted track to the desired new position.



**Figure 2 View of the genome browser in DeepView**

- e. A gene or locus can also be inputted in the “position” and “gene” textboxes. By clicking on the “jump” button, the desired locus is represented in full-size on the genome canvas.
- f. The “bookmark manager” button may be used to mark loci that you are interested in for further use.
- g. The “bookmark” button is used to select loci that you have marked previously.



**Figure 3 Zoom in by dragging the mouse**

- h. DNA and RNA buttons on the upper-right corner of the DeepView browser may be used to obtain DNA sequence and RNA secondary structures in the genome browser window. The resulting sequence will be displayed in a pop up window as shown in Figure 4. The structure picture is presented as the Vienna RNAfold application tool [2] with the secondary structure represented by brackets. The secondary structure figure is shown as well.

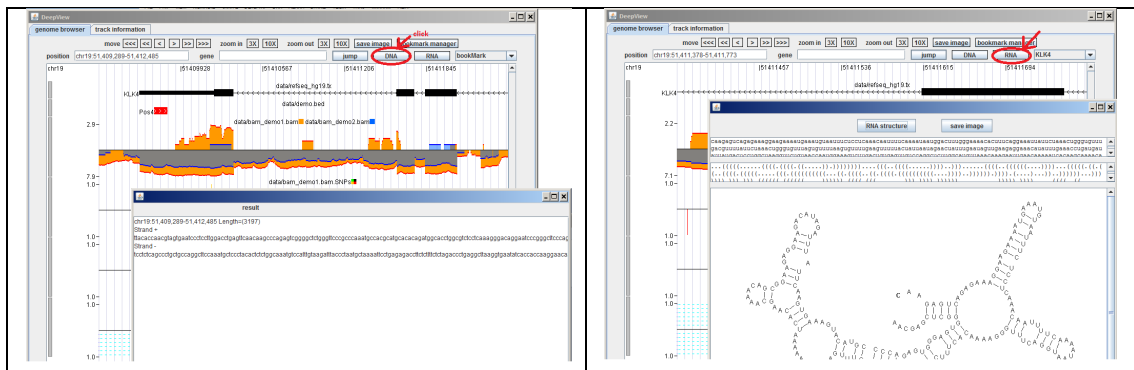


Figure 4 DNA and RNA secondary structure

## 1.3.2 Track Management

### 1.3.2.1 Adding Track

Users can add, remove, and edit tracks in a corresponding genome assembly using GUI as shown in Figure 5. In the download version, three genome assemblies were embedded (hg18, hg19, mm10). There are five items (Reference, Bed, UCSC wiggle, BAM wiggle and BAM reads) in the dropdown menu next to: “Add Track”. After selecting one item from the dropdown menu, an input dialogue pop-up appears (as shown in Figure 6).

Each Reference or Bed track corresponds to one data file. After choosing a Reference or Bed file and clicking “data file”, followed by “OK”, the track in XML format is appended in the textbox. The figure appears in “genome browser” tab, after clicking “refresh” button.

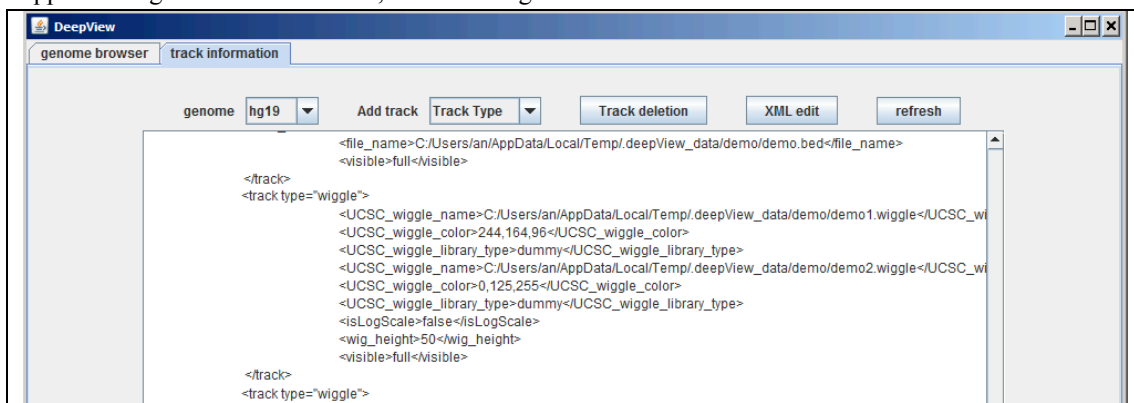


Figure 5 customize tracks

DeepView enables wiggle tracks with more than one wiggle file to be overlaid in a semi-transparent manner. In the wiggle track input dialogue, there is a “+” button to dynamically increase wiggle file. To compare wiggles, the representations of different wiggle files may be set to appear in contrasting colours. Similarly, this can be done for BAM wiggle file.

The only difference between two types of wiggle is that the UCSC wiggle track uses UCSC format, while BAM wiggle uses the indexed BAM file as input file.

The BAM track has three types: raw reads, SNP and InDel. The format of all their input files is indexed within the BAM file.

**Figure 6 track input interface**

**a. Reference track**

Users can add Reference tracks, which represent transcript annotation, such as Refseq, Ensembl, Genecode, and Aceview. These transcript files can be downloaded from UCSC genome browser web page. The files should have the extension “.tx”.

Figure 6 shows all input interfaces for the “add track” process. In general, you need to create a reference track from a transcript annotation for the target genome. This has two functions:

- To make a gene symbol query from the genome browser.
- To indicate genes present in the locus of the genome browser.

The reference track is described in track.xml as shown:

```
<track_reference>
  <file_name>geneCode_V12_hg19.tx</file_name>
  <visible>full</visible>
</track_reference>
```

The tag <file\_name> is followed by a transcript annotation file that is a text file with “.tx” as its extension. It is downloaded from UCSC genome browser. There are 15 columns, as listed below. The other annotation files can be downloaded from <http://genome.ucsc.edu/cgi-bin/hgTables>

- #bin
- Name
- Chrom
- Strand
- txStart
- txEnd
- cdsStart      cdsEnd
- exonCount
- exonStarts
- exonEnds
- score
- name2
- cdsStartStat
- cdsEndStat
- exonFrames

#### b. BED track

The UCSC genome browser BED and bigBed format are adopted.

Six, nine, and 12 column formats are supported in DeepView.

The first line contains column headings. Below is an example of the BED track with demo data:

chrom	chromStart	chromEnd	name	score	strand	thickStart	thickEnd	itemRgb	blockCount	blockSizes	blockStarts
chr19	51409723	51409828	Pos4	0	+	51409723	51409828	255,0,0			
chr19	51412723	51412838	Neg1	0	-	51412723	51412838	0,0,255			

The definition of the columns is the same as the UCSC genome browser, namely:

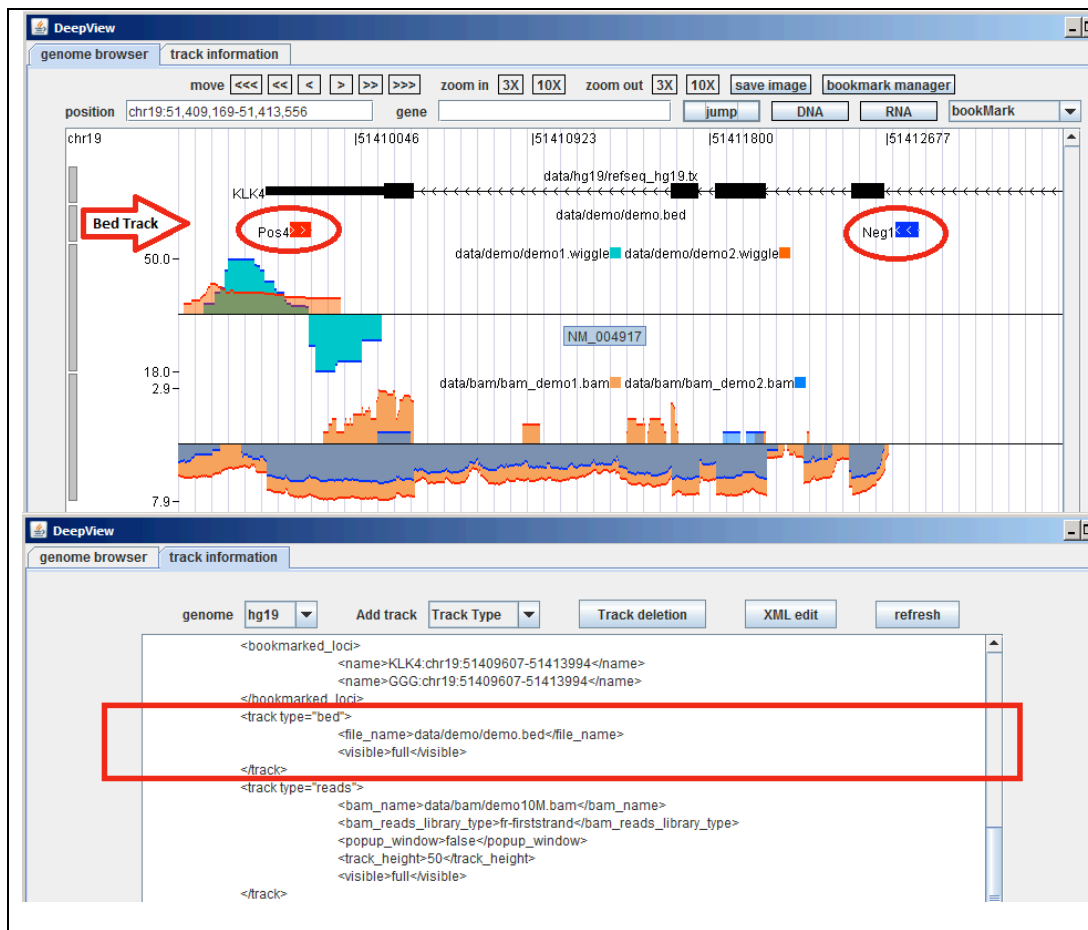
- chrom** - The name of the chromosome (e.g. chr7, chrX, chr2\_random).
- chromStart** - The starting locus of this item. The first base in a chromosome is numbered 0.
- chromEnd** - The end locus of this item. The chromEnd base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as chromStart=0, chromEnd=100, and the display spans the bases numbered 0 to 99.
- name** - The name of this item. This label is displayed at the top of the BED line in the genome browser canvas. When the mouse hovers over the item for a few seconds, the name will appear.
- score** – This is not used in the DeepView tool.
- strand** - Defines the strand as either '+' or '-'.

7. **thickStart** - The starting position at which the feature is drawn thickly (for example, the start exon in the gene display).
8. **thickEnd** - The end position at which the feature is drawn thickly (for example, the last exon in the gene display).
9. **itemRgb** - An RGB value of the form Red, Green, Blue (e.g. red colour: 255,0,0).
10. **blockCount** - The number of exons in the BED line.
11. **blockSizes** - A comma-separated list of the exon sizes. The number of items in this list should be the same as the blockCount.
12. **blockStarts** - A comma-separated list of block starts. All of the blockStart positions should be calculated relative to chromStart. The number of items in this list should be the same as blockCount.

The tag <file-name> is used for BED file name. In the XML file, <track\_bed> tag is used as follows:

```
<track_bed>
  <file_name>demo.bed</file_name>
  <visible>full</visible>
</track_bed>
```

Figure 7 shows how the BED track is shown in the genome browser and its setting in track.xml.



**Figure 7 BED custom track**

### c. Wiggle track

This is the same as the BedGraph and bigwig track format in the UCSC genome browser:

```
chr19 51409710 51409730 10
chr19 51409730 51409800 8
```



*chr19 51409800 51409820 7*  
*chr19 51402000 51402050 -10*  
*chr19 51402050 51402150 -18*

There are four columns in the wiggle track:

1. **chrom** - The name of the chromosome (e.g. chr7, chrX, chr2\_random).
2. **chromStart** - The start of the locus for this item. The first base in a chromosome is numbered 0.
3. **chromEnd** - The end of the locus for this item. The chromEnd base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as chromStart=0, chromEnd=100, and the display spans the bases numbered 0 to 99.
4. **value** - the expression level. This corresponds to the height of the bar in the genome browser canvas. Positive values are represented by “+” strand and negative values are represented by “-”. These will be displayed in the genome browser canvas when the mouse pointer hovers over this position.

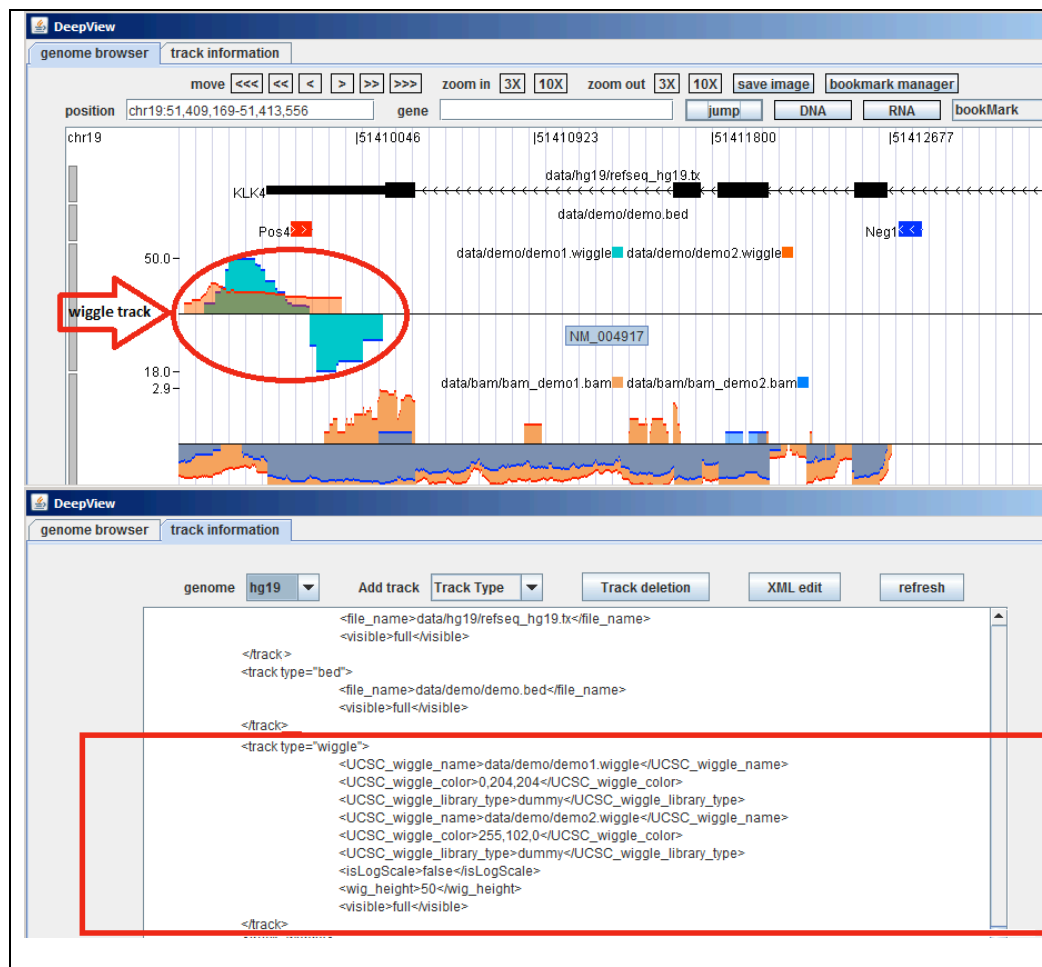
Figure 8 shows two wiggle files are semi-transparently overlaid into one track. The description to show overlaid wiggle files is as follows:

```
<track_wiggle>  
  <UCSC_wiggle_name>demo1.wig</UCSC_wiggle_name>  
  <UCSC_wiggle_color>244,164,96</UCSC_wiggle_color>  
  <UCSC_wiggle_name>demo2.wig</UCSC_wiggle_name>  
  <UCSC_wiggle_color>0,125,255</UCSC_wiggle_color>  
  <isLogScale>>false</isLogScale>  
  <wig_height>50</wig_height>  
  <visible>full</visible>  
</track_wiggle>
```

The tag <UCSC\_wiggle\_name> encloses the wiggle file name. The colour RGB 244,164,96 is given for the wiggle. The second wiggle file is described in the same way.

If tag <isLogScale> is set as “true”, the value lies in the natural log scale. Otherwise the original value is used.

The tag <wig\_height> shows the height of the track. The tag <visible> is set to “full”.



**Figure 8 How to add a wiggle track.**

The top track of the genome browser canvas shows the added demo\_wiggle track. Mousing over on the edges of the wiggle shows the value of a wiggle item.

#### d. BAM track

DeepView produces three types of track for the BAM file:

##### 1. Wiggle (coverage)

The genome coverage is calculated from this BAM file.

In the XML file, the BAM file name is set by tag `<bam_wiggle_name>`. The colour is set by tag `<bam_wiggle_color>`. The two tags can be repeated multiple times.

The wiggles generated by these BAM files will be overlaid as wiggle tracks described in Section 2.3.2. The tag `<isLogScale>` determines whether the values lie in the natural logscale. The remaining two tags `<wig_height>` and `<visible>` are the same as for the wiggle track.

Figure 9 shows the two BAM files `bam_demo1.bam` and `bam_demo2.bam`.

```
<track_wiggle>
  <bam_wiggle_name>bam_demo1.bam</bam_wiggle_name>
  <bam_wiggle_color>244,164,96</bam_wiggle_color>
  <bam_wiggle_name>bam_demo2.bam</bam_wiggle_name>
  <bam_wiggle_color>0,125,255</bam_wiggle_color>
  <isLogScale>true</isLogScale>
  <wig_height>50</wig_height>
  <visible>full</visible>
</track_wiggle>
```

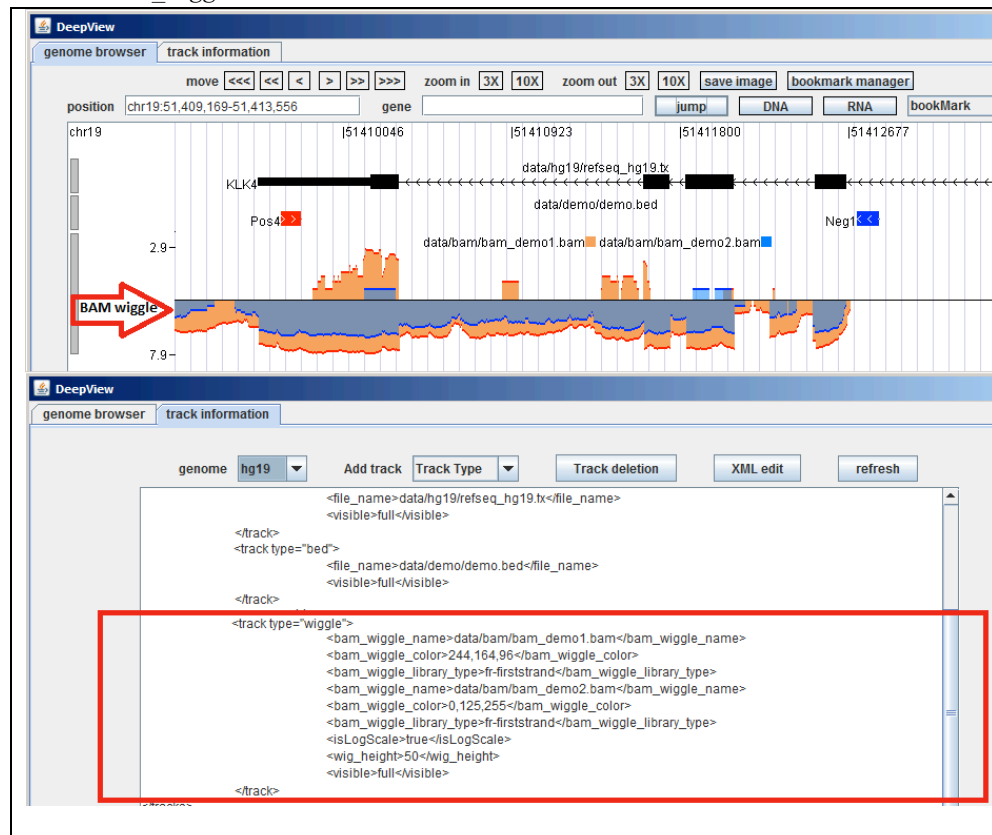
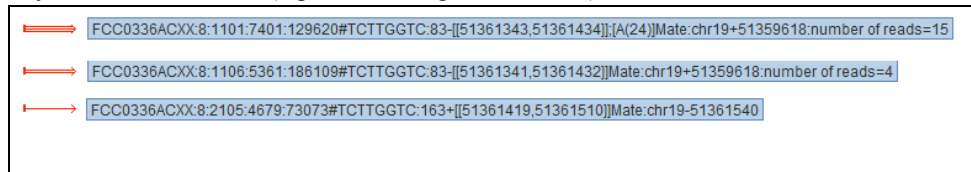


Figure 9 Two demo BAM files comprising a `bam_wiggle` track.

##### 2. Raw reads

The raw reads track shows all reads aligned in the genome – using arrows to indicate read direction.

Red arrows indicate that the read’s direction is from left to right; while blue arrows show that the read’s direction is from right to left. For properly mapped, paired reads, one read should be from left to right and other should be from right to left in Transcript direction. In order to reduce memory requirements and to compress the pop-up window for the ‘raw reads’ track, paired reads that have the exact sequence are only shown once. However, the thickness of the arrow represents how many reads are represented for those multiple reads. For example, 1 to 2 reads are represented with 1 line in the arrow (bottom arrow in Figure 10 below), while 3 to 10 reads are represented by 2 lines in the arrow (middle arrow in Figure 10 below), and more than 10 reads are represented by 3 lines in the arrow (top arrow in Figure 10 below).



**Figure 10 mark of reads**

The green solid line connects the paired reads that are properly mapped. The green dotted line indicates that a read is split in the genome.

The track height can be selected in the input interface. Since the number of raw reads mapping to the same chromosome loci usually amount to several hundreds, (sometimes, even tens of thousands), DeepView provides a scrollbar on the left side of track. For extreme high expressed region, user can make a popup window with scrollbar, which allows the user to check every single read as shown in Figure 12. The description for raw reads is:

```
<track_reads>
  <bam_name>data/bam_demo1.bam</bam_name>
  <bam_reads_library_type>fr-firststrand</bam_reads_library_type>
  <popup_window>true</popup_window>
  <track_height>50</track_height>
  <visible>full</visible>
</track_reads>
```

The library type is used to determine the transcript direction. Track height determines how many reads will be displayed in the track. Every read has a “mouse-over” display function. The display consists of read name, SAM flag, and its alignment loci followed by mate read coordinates.

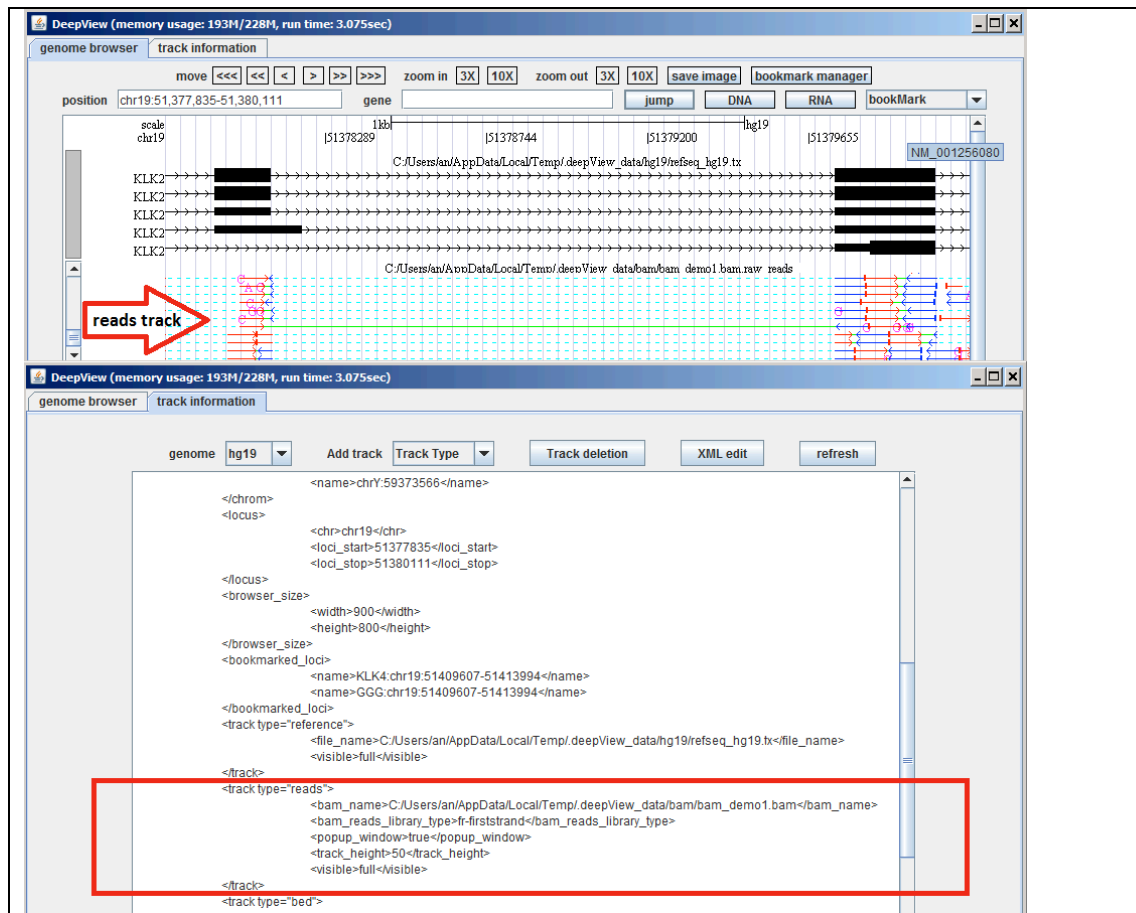


Figure 11 Raw reads plot

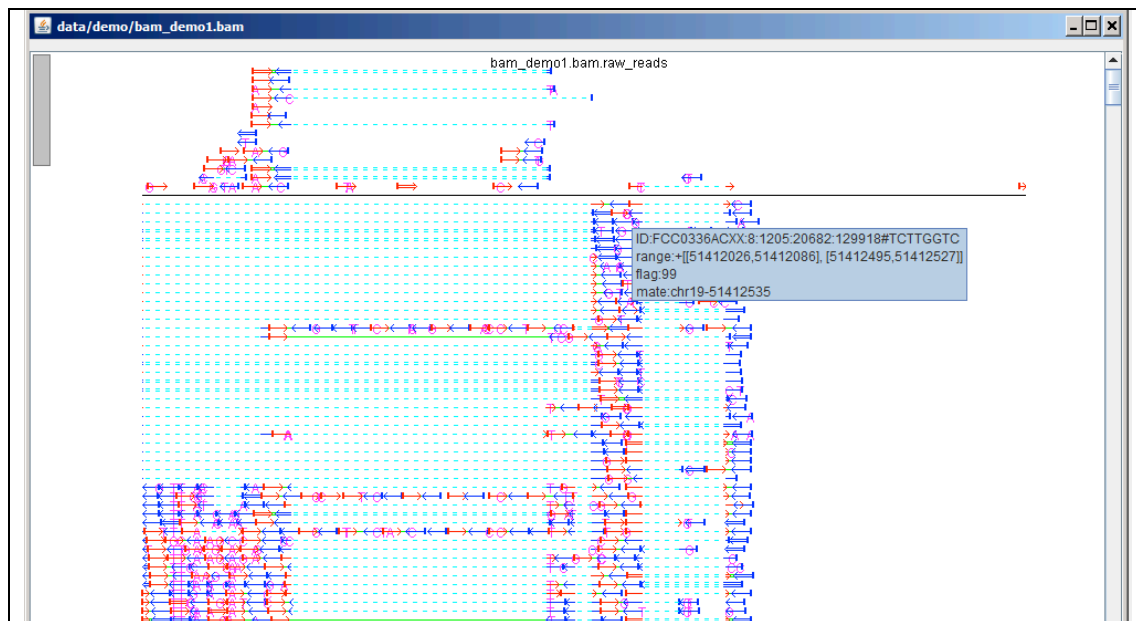


Figure 12 raw reads window

### 3. SNP track (mismatch of single nucleotide) (SNPs)

SNP information is extracted from the “MD” attribute in the BAM file.

Bars with different colours denoting different genomic nucleotides (for example: “A”: green, “C”: black, “G”: yellow, and “T”: red) are displayed in the SNP positions.

The determination of SNP or mismatch is based on the individual expression of read; therefore, most SNPs only appear in “+” or “-” strands because most of the genome regions are only expressions of one strand. The bam\_SNP track is enclosed by the tag <bam\_SNP\_name>.

Figure 13 shows SNPs for BAM file bam\_demo1. The description of the SNP track is

```
<track_SNP>
    <bam_name>data/bam_demo1.bam</bam_name>
    <min_total_coverage>8</min_total_coverage>
    <min_variant_coverage>2</min_variant_coverage>
    <min_VarFreq>0.2</min_VarFreq>
    <min_avgQuality>15</min_avgQuality>
    <max_pval>0.01</max_pval>
    <bam_SNP_library_type>fr-firststrand</bam_SNP_library_type>
    <wig_height>50</wig_height>
    <visible>full</visible>
</track_SNP>
```

The SNP calling uses VarScan algorithm [8, 9].

Where the item “min\_total\_coverage” represents the minimum read depth of the SNP variant allele.

“min\_variant\_coverage” represents the minimum number of reads with variant allele.

“min\_VarFreq” represent s the minimum ratio of the number of reads with variant allele to the number of reads with reference genome allele.

“min\_argQuality” represents the min average base quality of the reads.

“max\_pval” represents the maximum fisher exact distribution p-value for variant allele comparing to genome reference allele.

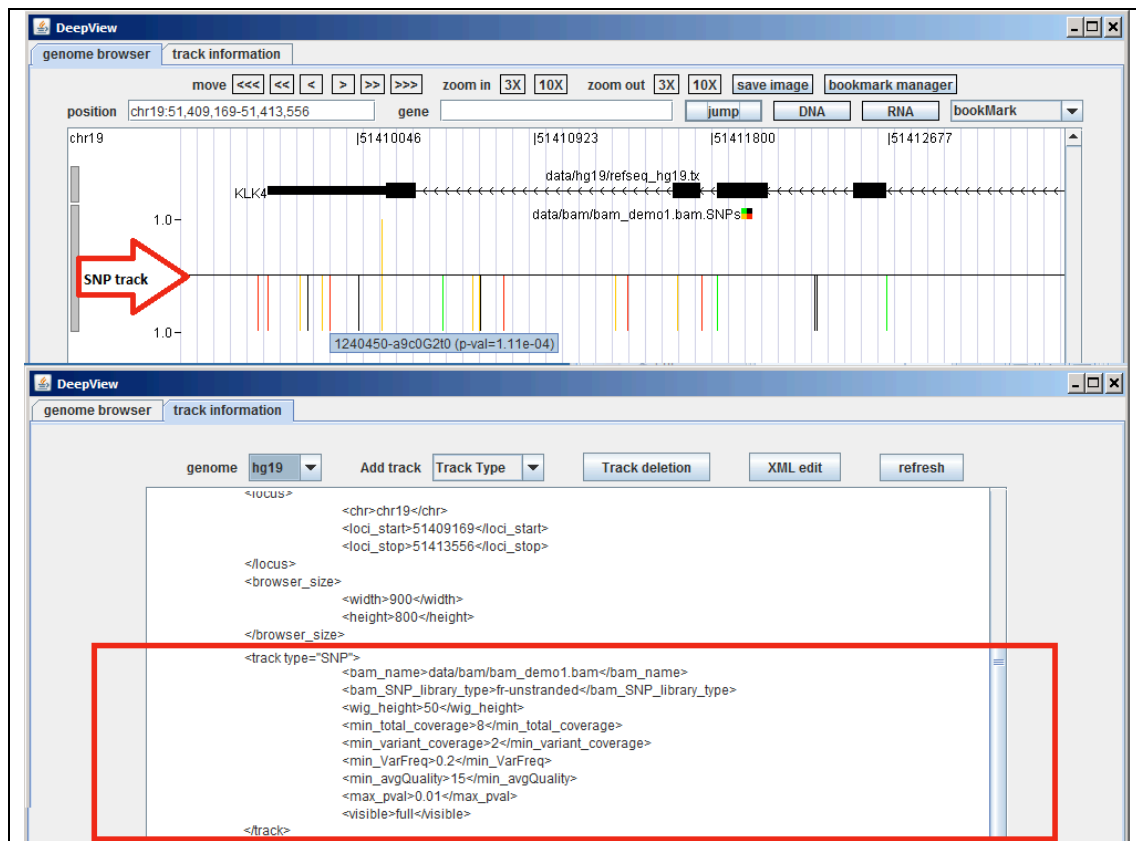


Figure 13 SNPs for BAM file of a demo bam\_SNP Track

#### 4. InDel track (insertion and deletion)

The bam\_InDel track is similar to the bam\_SNP track. Insertion and deletion information is gathered from the Cigar flag and “MD” attributes of BAM files.

The tag <bam\_InDel\_name> encloses the BAM file from which the InDel information is extracted. The bar corresponding to the insertion position is **red**, and the bar corresponding to the deletion position is **blue**. The description in XML for the InDel is:

```
<track_InDel>
    <bam_name>data/bam_demo1.bam</bam_name>
    <bam_InDel_library_type>fr-firststrand</bam_InDel_library_type>
    <wig_height>50</wig_height>
    <visible>full</visible>
</track_InDel>
```

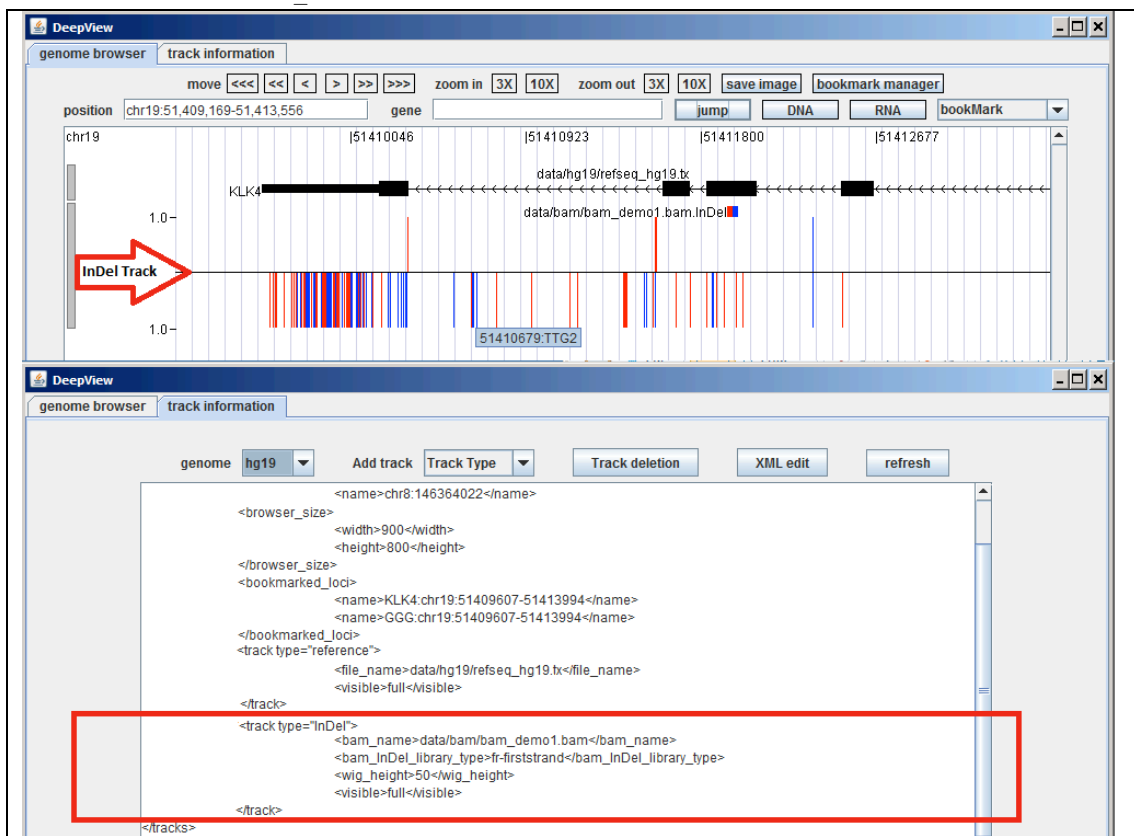


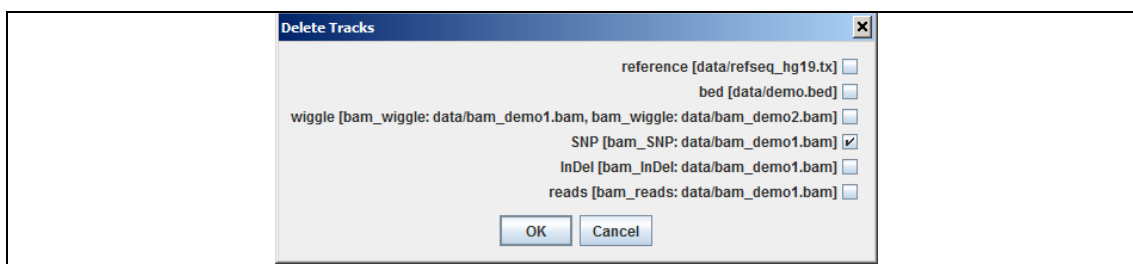
Figure 14 A BAM\_InDel track

#### 1.3.2.2 Deleting Track

A window showing the inputted tracks pops up when clicking “Track deletion” (popup shown in Figure 5). For Reference and Bed track types the listings in the popup window each start with track type, followed by the track file name. For other track types, the subtype of track appears before the track file name.

Users can select the track type to be deleted by checking the box next to the relevant track name in the popup and then clicking the OK button.

If you want to restore the deletion, you need to click “refresh” to reload tracks.



**Figure 15 track deletion**

### 1.3.2.3 Editing track information

The textbox is not editable before clicking “XML edit” (shown in Figure 5).

Each track corresponds to one closed XML item between <track\_xxx> to </track\_xxx>.

If a user is familiar with XML format, user can modify the XML text file directory without using the pop up dialogue box (shown in Figure 6). However, this should be done very careful, as the XML file may crash and DeepView will not start up successfully, because of small typo or wrong XML format.

## References

1. Fujita P.A., Rhead B., Zweig A.S., Hinrichs A.S., Karolchik D., Cline M.S., Goldman M., Barber G.P., Clawson H., Coelho A., et al. The UCSC Genome Browser database: update 2011. *Nucleic Acids Res.* 2011;39:D876-D882.
2. I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoeffer, M. Tacker, P. Schuster (1994) Fast Folding and Comparison of RNA Secondary Structures. *Monatshefte f. Chemie* 125: 167-188.
3. Karolchik,D., Baertsch,R., Diekhans,M., Furey,T.S., Hinrichs,A., Lu,Y.T., Roskin,K.M., Schwartz,M., Sugnet,C.W., Thomas,D.J. et al. (2003) The UCSC Genome Browser Database. *Nucleic Acids Res.*, 31, 51–54.
4. Rhead,B., Karolchik,D., Kuhn,R.M., Hinrichs,A.S., Zweig,A.S., Fujita,P.A., Diekhans,M., Smith,K.E., Rosenbloom,K.R., Raney,B.J. et al. (2010) The UCSC Genome Browser database: update 2010. *Nucleic Acids Res.*, 38, D613–D619.
5. Li H, Handsaker B, Wysoker A, et al. The sequence alignment/map format and SAMtools. *Bioinformatics* 2009;25:2078-9.
6. Lai J, Lehman ML, Dinger ME, Hendy SC, Mercer TR, Seim I, Lawrence MG, Mattick JS, Clements JA, Nelson CC. A variant of the KLK4 gene is expressed as a cis sense-antisense chimeric transcript in prostate cancer cells. *RNA* 2010, 16:1156–1166.
7. James T. Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, Jill P. Mesirov. Integrative Genomics Viewer. *Nature Biotechnology* 29, 24–26 (2011).
8. Koboldt DC, Chen K, Wylie T, Larson DE, McLellan MD, Mardis ER, Weinstock GM, Wilson RK, & Ding L (2009). VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* (Oxford, England), 25 (17), 2283-5 PMID: 19542151
9. VarScan 2: Koboldt, D., Zhang, Q., Larson, D., Shen, D., McLellan, M., Lin, L., Miller, C., Mardis, E., Ding, L., & Wilson, R. (2012). VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing *Genome Research* DOI: 10.1101/gr.129684.111