Hpo Case Annotator Documentation

Release 2.0.0

Daniel Danis, Peter N Robinson

Apr 12, 2023

Index:

Hpo Case Annotator is an application for biocuration of case reports, families and cohorts of patients published in scientific literature. Each curated case contains details of the disease-causing variants, phenotype, disease, and other metadata.

Curated data is stored in *JSON* format (one file for each case). The application also performs a number of Q/C checks to ensure data consistency.

The app can export data in Phenopacket format, but it contains a superset of the information required for phenopackets. Future versions of this app will probably converge to the Phenopacket format, and currently the app is still in a preliminary stage of development, although it works as advertised.

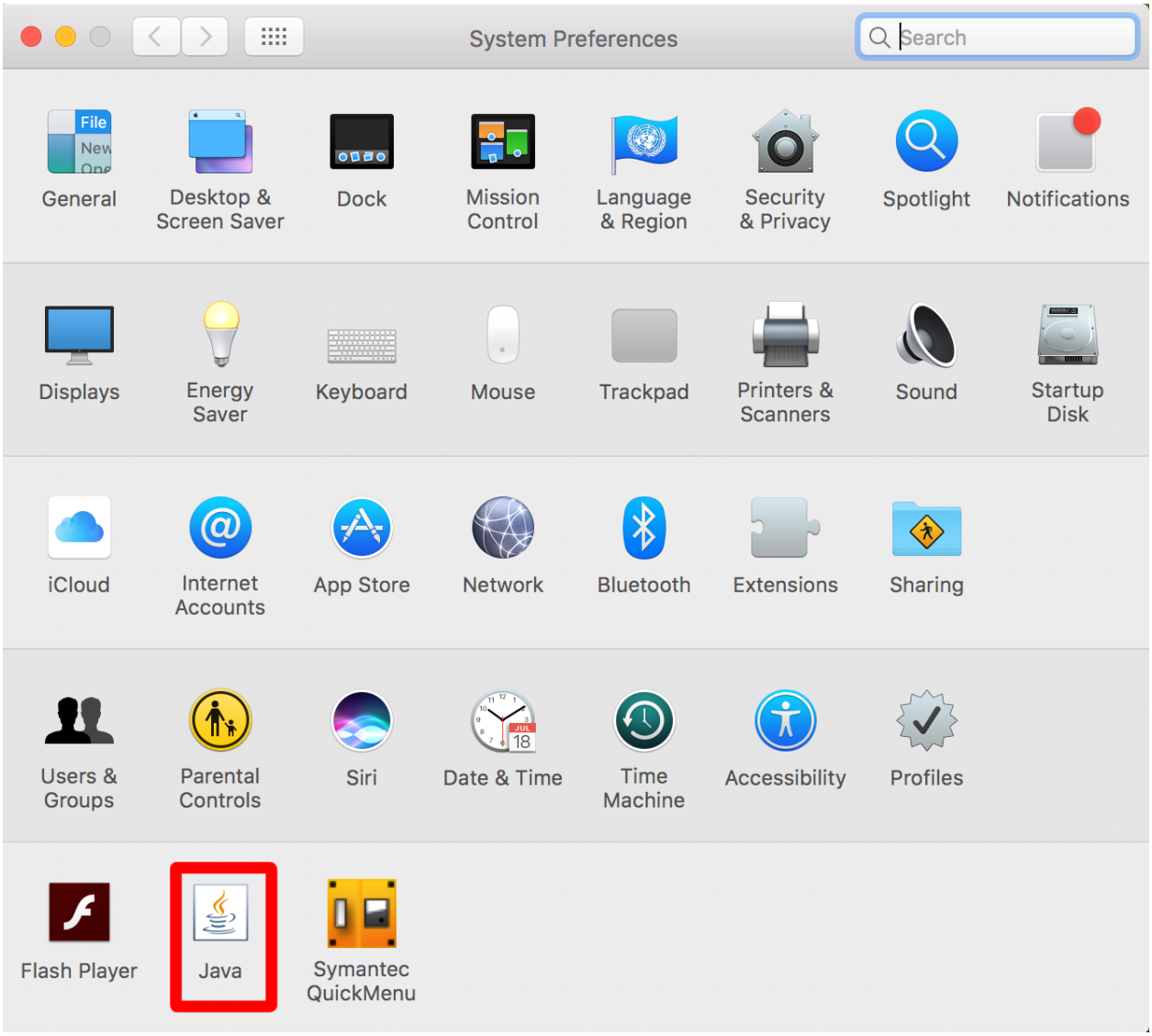
CHAPTER 1

Requirements

Hpo Case Annotator is a Java app and it requires **Java 17** or better to be installed in the environment. This page describes steps required to check which version of Java (if any) is installed on your *Mac*, *Linux* or *Windows* machine.

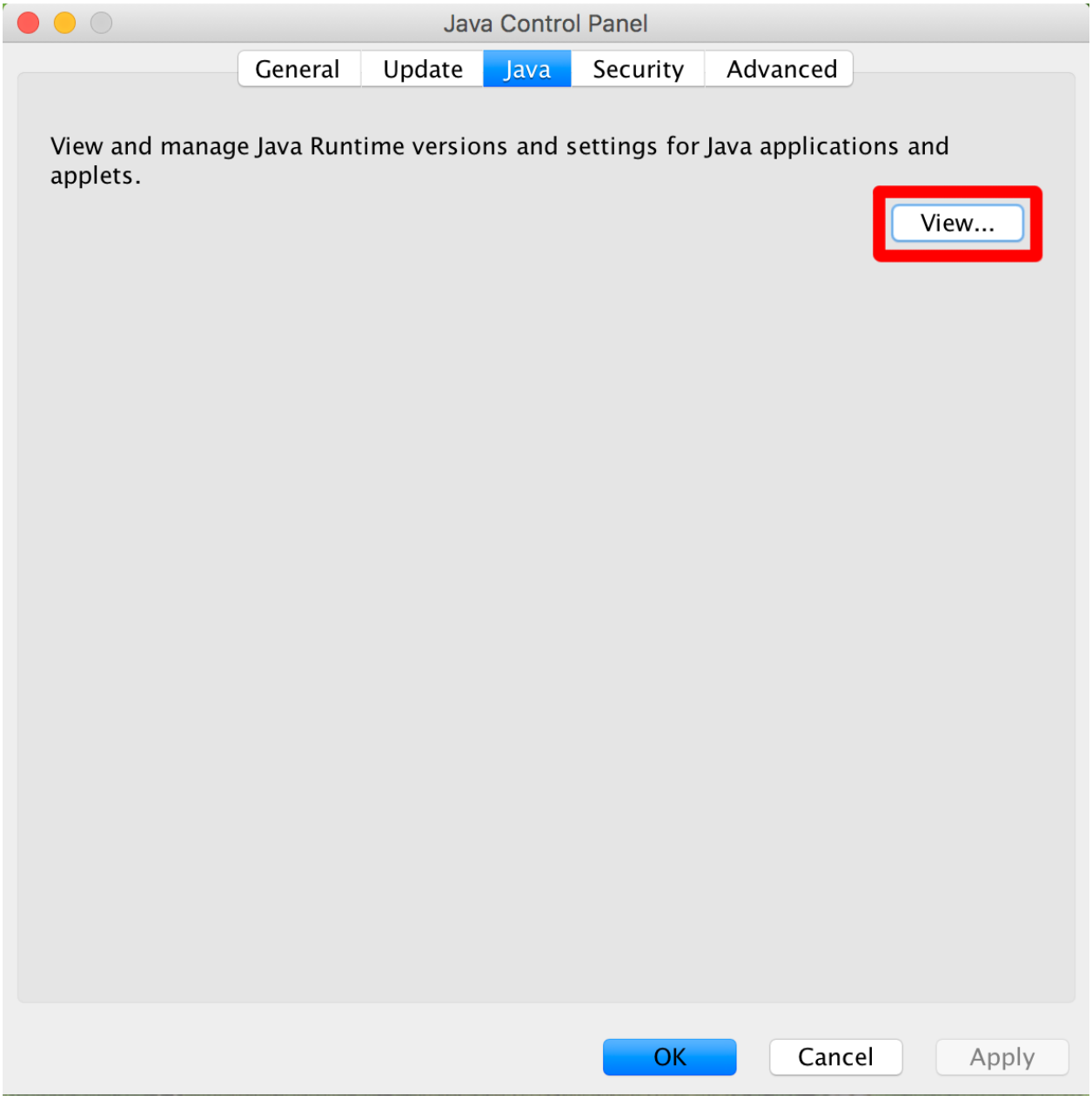
1.1 Mac OSX

Open the **System preferences** and click on Java.

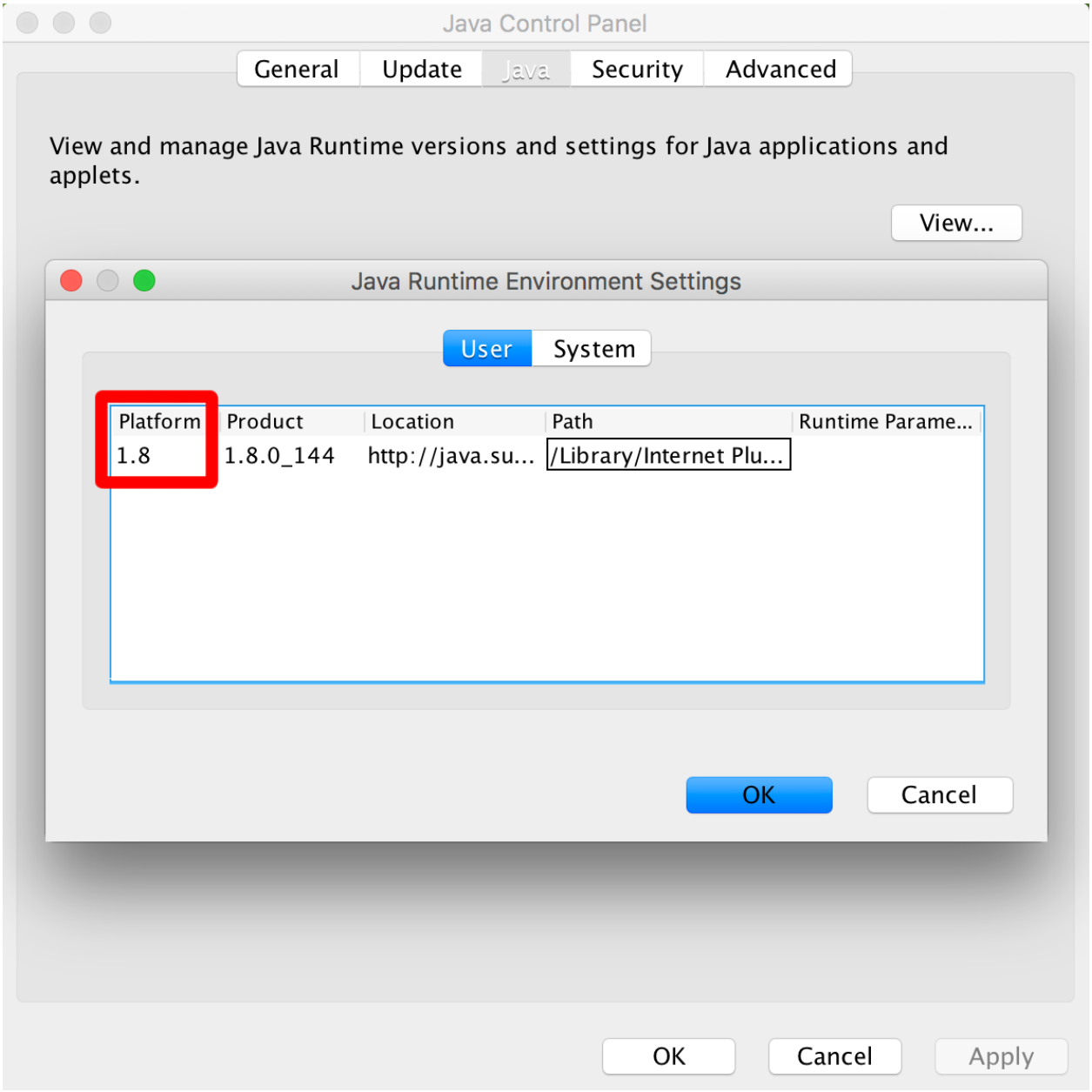


Select the **Java** tab at the top of the window

Click on the **View** button.



You should see *17* or better (instead of *1.8*) in the *Platform* column of the table.



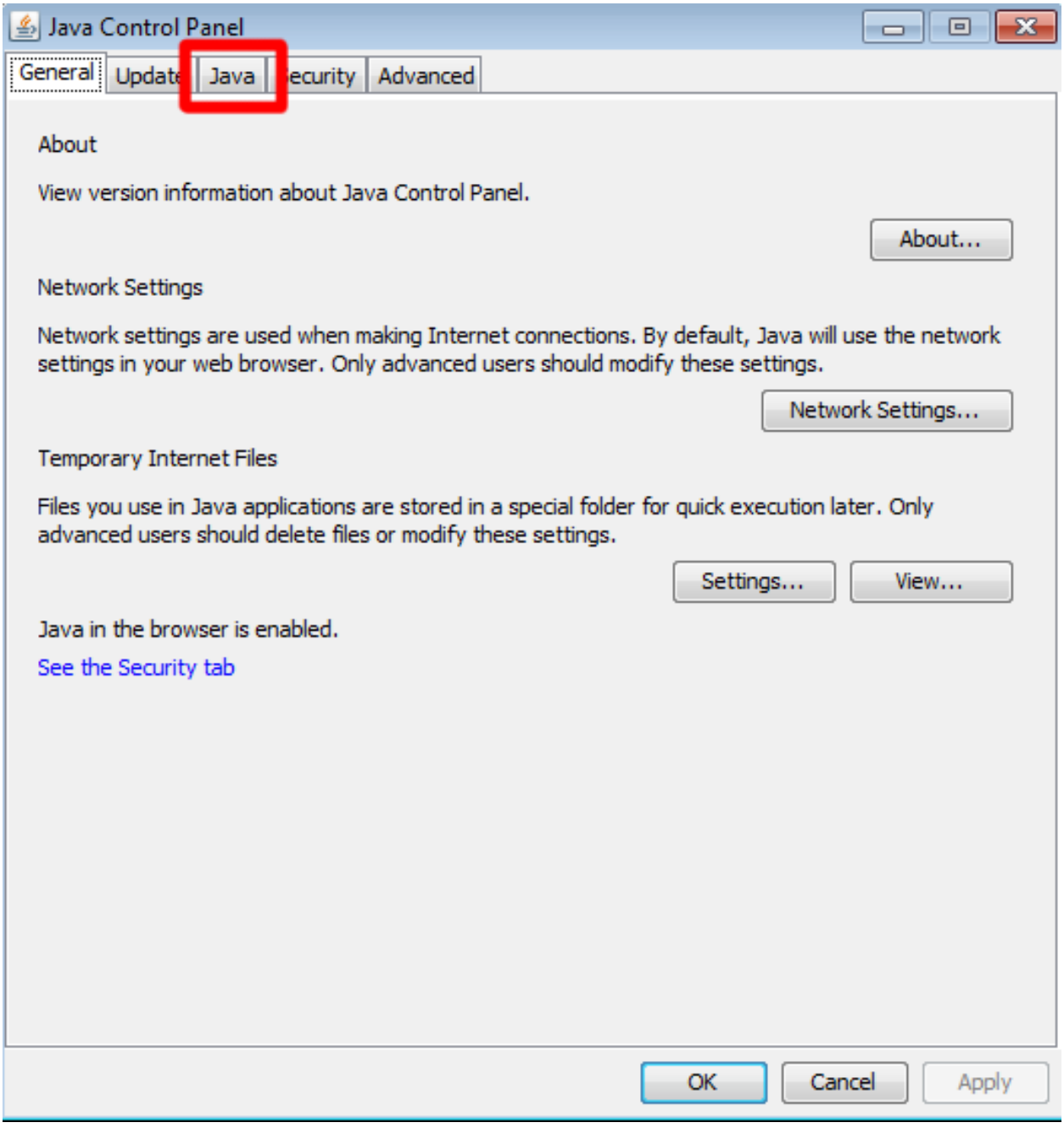
1.2 Linux

You can determine what version of **Java** you have on your computer by running `java -version` in your Terminal:

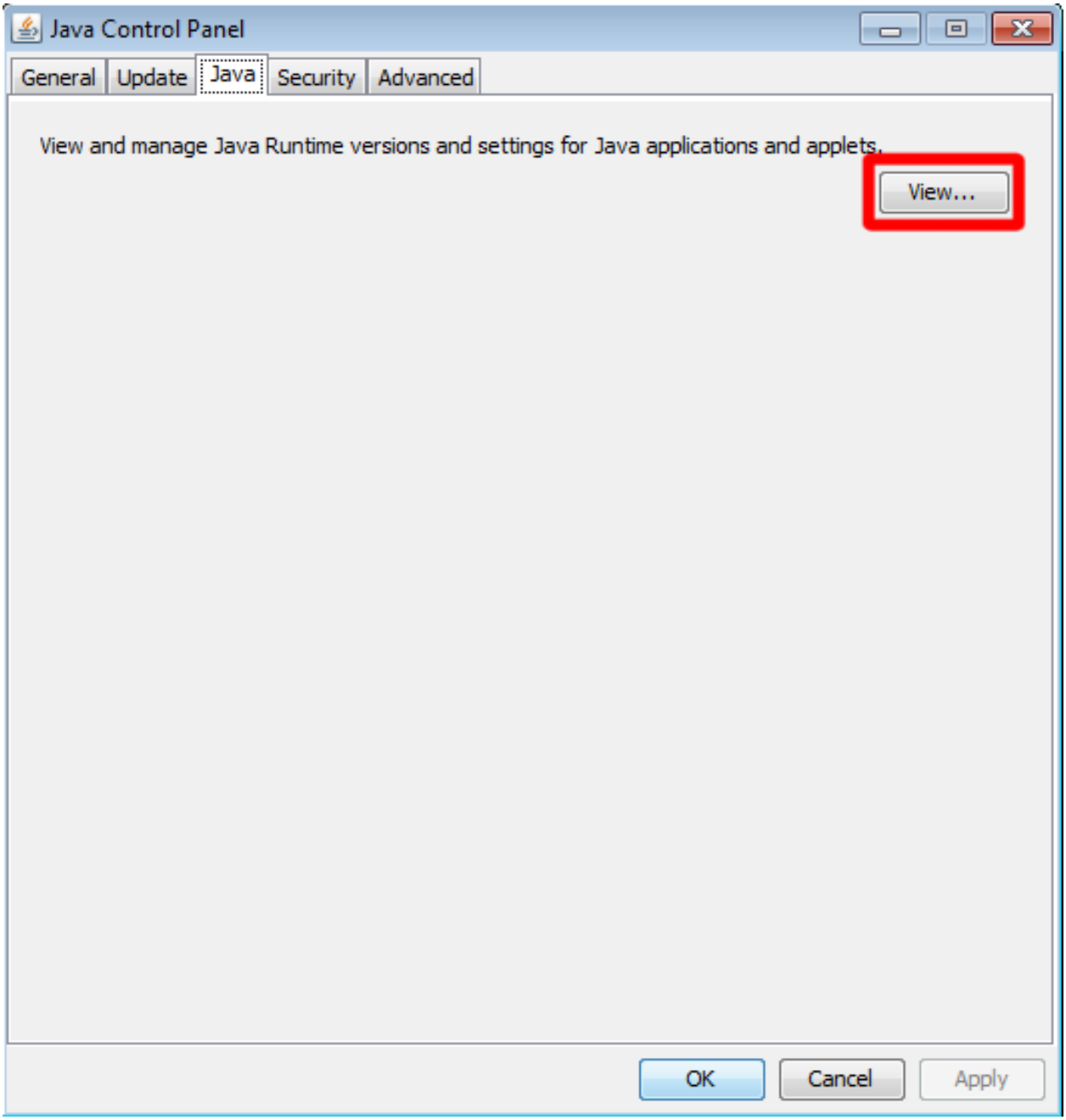
```
$ java -version
openjdk version "17" 2021-09-14
OpenJDK Runtime Environment (build 17+35-2724)
OpenJDK 64-Bit Server VM (build 17+35-2724, mixed mode, sharing)
```

1.3 Windows

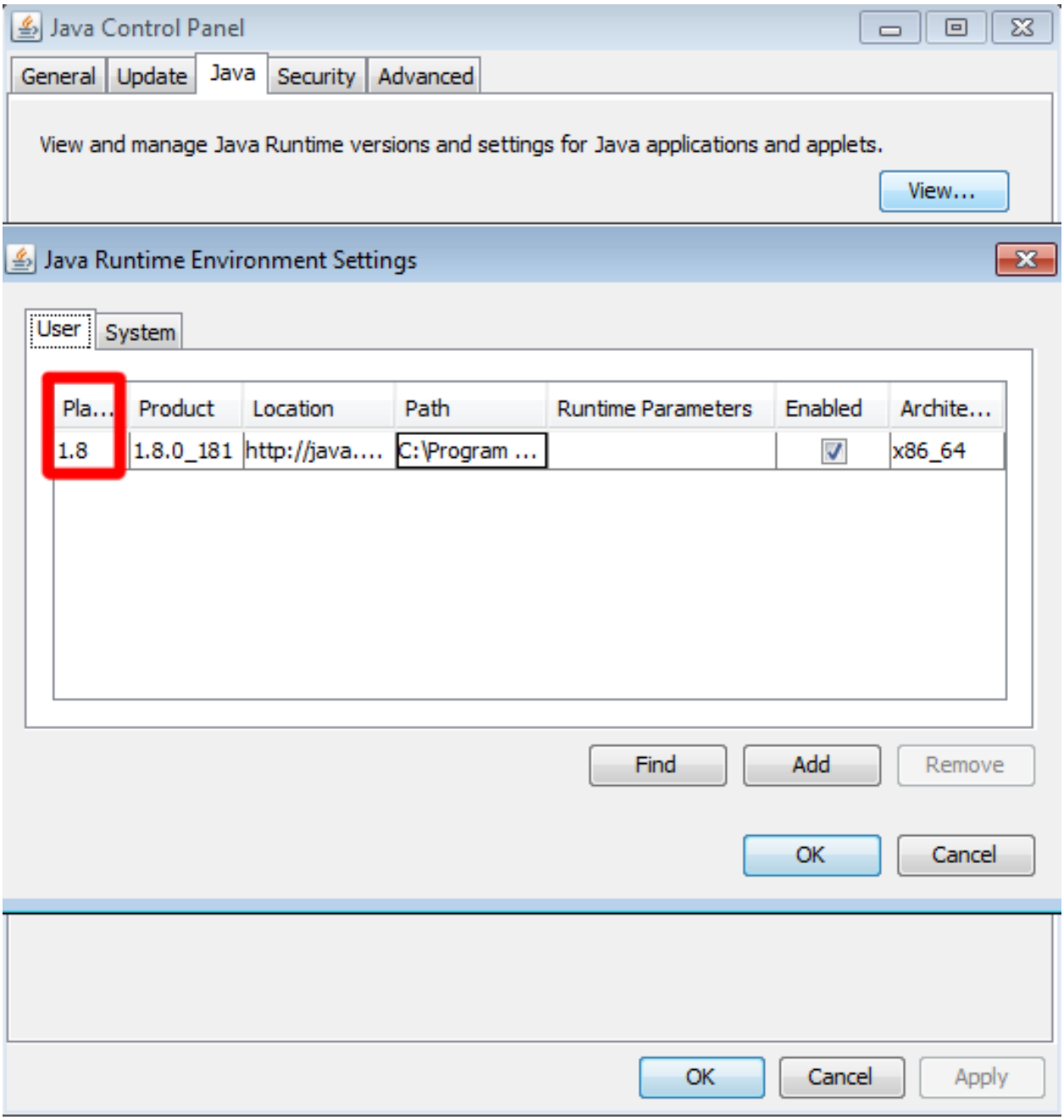
Open **Java Control Panel** and select the **Java** tab at the top of the window.



Click on the **View** button.



You should see *17* or better (instead of *1.8*) the *Platform* column of the table.



Having Java set up, let's move to the next step - setting up *HpoCaseAnnotator* on your machine.

CHAPTER 2

Setup

This document will guide you through setting up *HpoCaseAnnotator* on your machine. The setup consists of two steps:

1. getting the app (prebuilt archive or building from sources)
2. setting up the resources

2.1 Get HpoCaseAnnotator

2.1.1 Prebuilt app

Most users (Mac, Linux, Windows) should download the distribution ZIP archive available at HpoCaseAnnotator releases. Make sure you download the ZIP for your platform and unpack the archive.

2.1.2 Build from sources

HpoCaseAnnotator can also be built from sources (Mac and Linux users).

First, we clone the repo from GitHub, and then use the amazing Maven wrapper to build the app:

```
$ git clone https://github.com/monarch-initiative/HpoCaseAnnotator.git
$ cd HpoCaseAnnotator
$ ./mvnw -Prelease package
```

Note: The build requires a working internet connection for downloading required libraries and Java Development Kit (JDK) 17 or better.

The build creates the distribution ZIP archive in the `hpo-case-annotator-app/target` folder.

2.1.3 Launch

HpoCaseAnnotator is started by double-clicking on a launcher script that is bundled in the distribution ZIP. The app ships with three launchers, one script per *Mac*, *Linux* and *Windows* platforms:

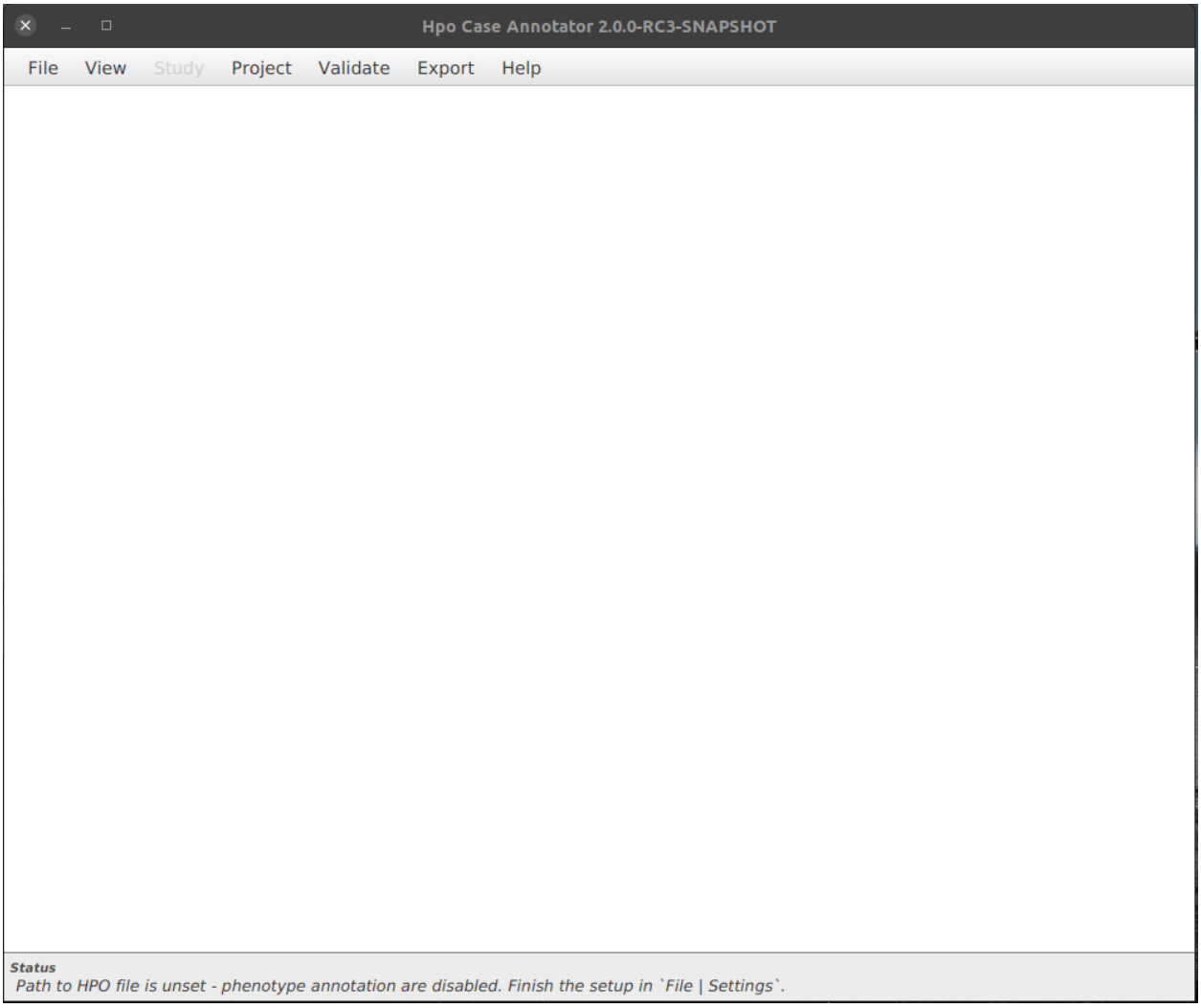
- **Mac** - open Finder and double-click on `launch.command`
- **Linux** - open file browser and double-click on `launch.sh`
- **Windows** - open Explorer and double-click on `launch.bat`

Note: You need to have Java Runtime Environment (JRE) 17 or better on your machine. See *Requirements* section for more info.

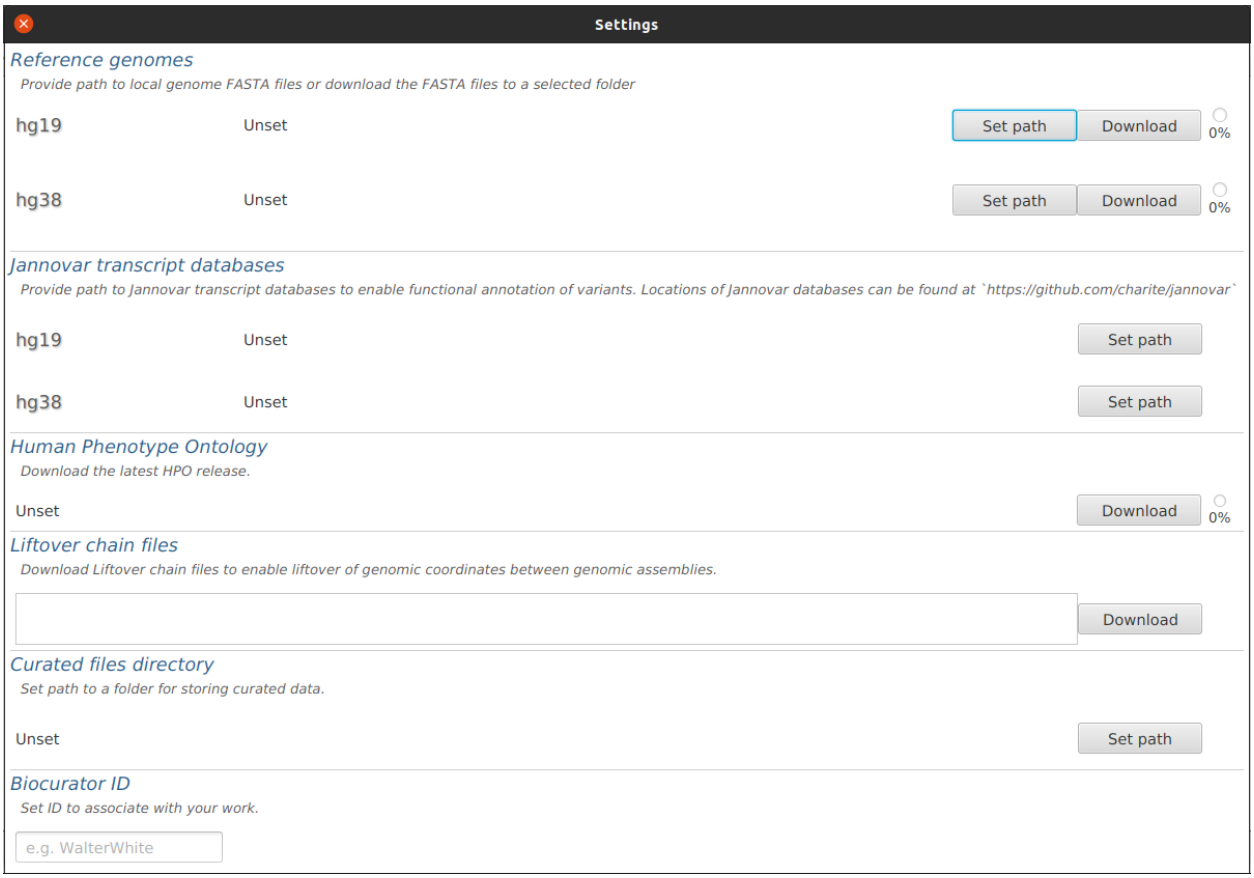
The app window will appear shortly after double-click.

2.2 Setup

Note, that *not* all the functionality is enabled after the first startup; the status bar in the bottom part of the screen indicates that e.g. path to HPO file is unset.



Go to `File | Settings` as directed - a new dialog window is opened:



Note that most of the resources are *Unset* or empty, we will fill the fields shortly.

2.2.1 Reference genomes

HpoCaseAnnotator needs access to the sequence of the reference genome to e.g. check if the wildtype sequence entered for each variant matches the corresponding genomic position. You can provide a local FASTA file yourself (*Set path* button) or *HpoCaseAnnotator* can download and pre-process the reference genome automatically (*Download* button).

Currently, **GRCh37 (hg19)** and **GRCh38 (hg38)** genome assemblies are supported. This is all we need for the Q/C routines.

Note: The reference genome files have ~4GB each. Handle with care.

2.2.2 Jannovar transcript databases

HpoCaseAnnotator uses Jannovar to perform functional annotation of variants with respect to genes and transcripts. Therefore, the app needs to know the location of Jannovar transcript databases. As of now, the databases must be downloaded manually. The download links are in Jannovar code repository.

Download databases for *H. sapiens* to a location of your choice. Both *ENSEMBL* or *RefSeq* will work fine (although only one can be used at the time). After download, click on *Set path* buttons to set paths.

2.2.3 Human Phenotype Ontology

HpoCaseAnnotator can download the latest version of *Human Phenotype Ontology* (HPO) in *JSON* format. The *JSON* file (~25 MB) is downloaded into *HpoCaseAnnotator* data folder which is located in your home directory. The download needs to be done once (and can be updated as necessary).

Click on *Download* button to download the *JSON* file.

2.2.4 Liftover chain files

HpoCaseAnnotator needs Liftover chain files to provide the liftover functionality. The chains for converting genomic positions from *hg18* or *hg19* to *hg38* (<1MB) are downloaded into *HpoCaseAnnotator* data folder after clicking on *Download* button.

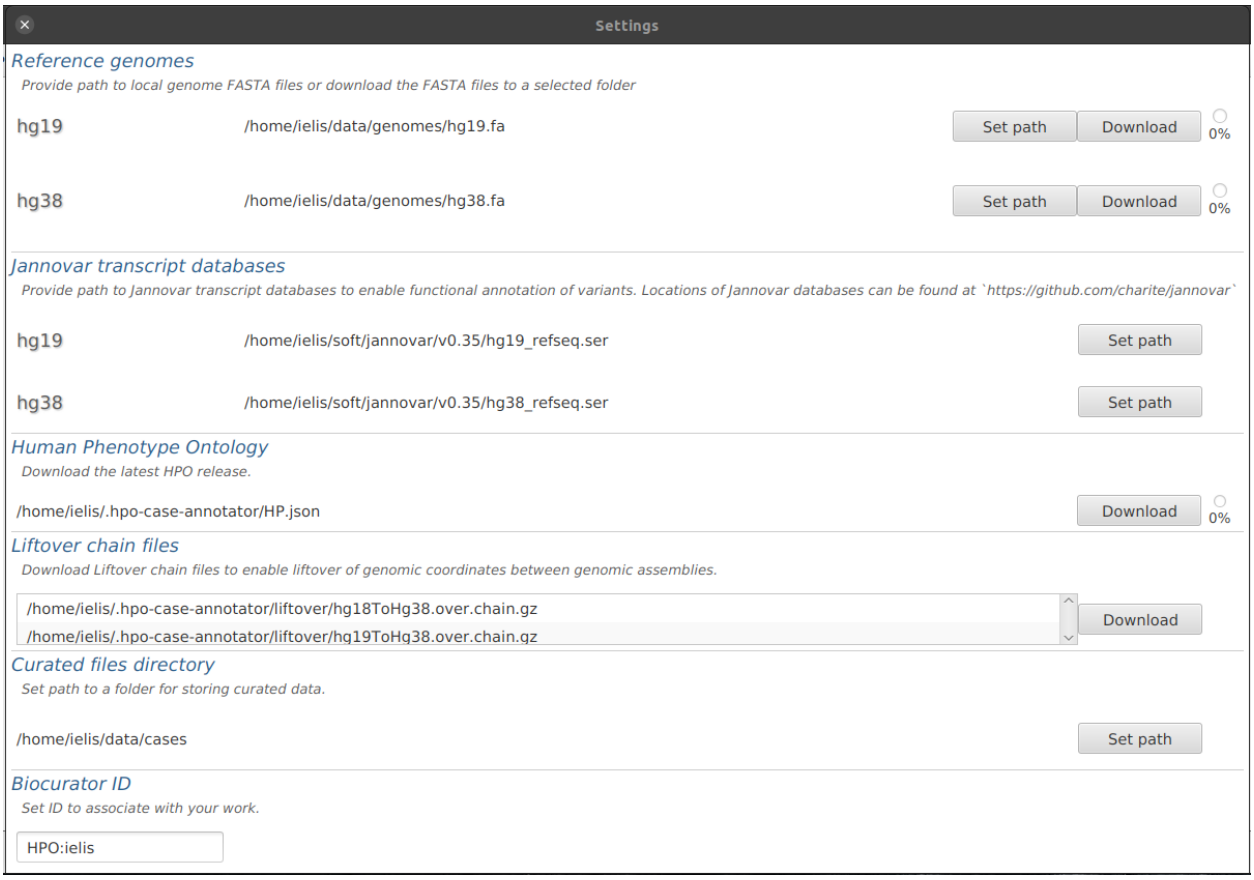
2.2.5 Curated files directory

Each curated case is stored as a *JSON* file. Here we set path to a directory where the *JSON* files are stored by default. We recommend using a directory per project.

2.2.6 Biocurator ID

Here provide your biocurator ID.

This setup and the resource download is done only once. And after these steps, the *Settings* dialog can be closed and *HpoCaseAnnotator* is fully prepared for work.


Settings
Reference genomes
Provide path to local genome FASTA files or download the FASTA files to a selected folder
hg19
/home/ielis/data/genomes/hg19.fa
Set path
Download
0%
hg38
/home/ielis/data/genomes/hg38.fa
Set path
Download
0%
Jannovar transcript databases
Provide path to Jannovar transcript databases to enable functional annotation of variants. Locations of Jannovar databases can be found at `https://github.com/charite/jannovar`
hg19
/home/ielis/soft/jannovar/v0.35/hg19_refseq.ser
Set path
hg38
/home/ielis/soft/jannovar/v0.35/hg38_refseq.ser
Set path
Human Phenotype Ontology
Download the latest HPO release.
/home/ielis/.hpo-case-annotator/HP.json
Download
0%
Liftover chain files
Download Liftover chain files to enable liftover of genomic coordinates between genomic assemblies.
/home/ielis/.hpo-case-annotator/liftover/hg18ToHg38.over.chain.gz
/home/ielis/.hpo-case-annotator/liftover/hg19ToHg38.over.chain.gz
Download
Curated files directory
Set path to a folder for storing curated data.
/home/ielis/data/cases
Set path
Biocurator ID
Set ID to associate with your work.
HPO:ielis

CHAPTER 3

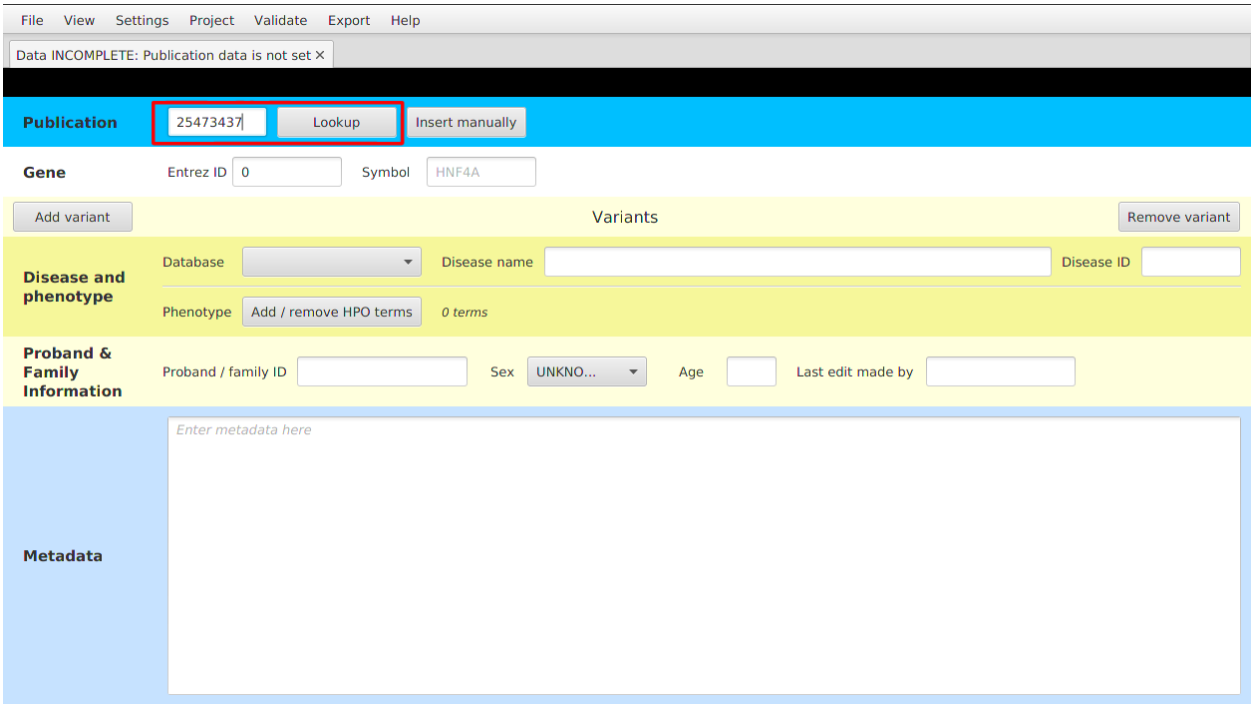
Entering data

Warning: The documentation has not been updated for the *2.** version yet.

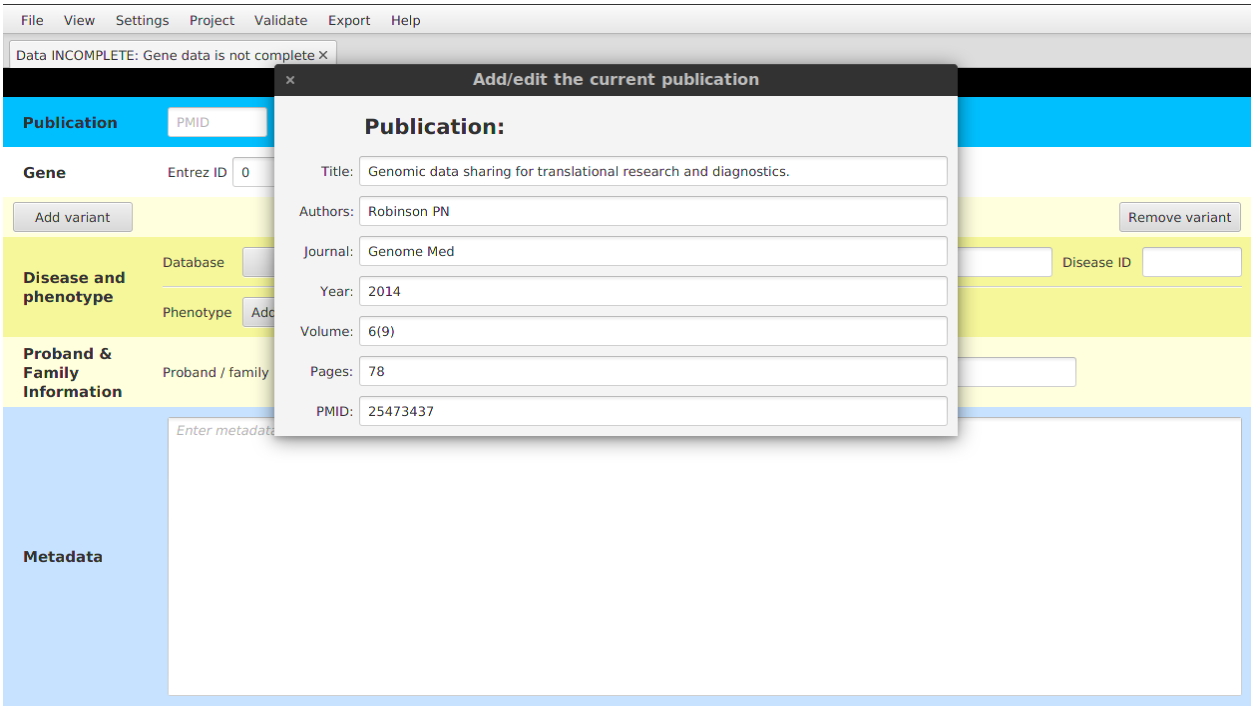
3.1 Publication

There are two ways of entering the data regarding the publication which describes the curated case:

1. *Using PMID* - enter the *PMID* number of the publication and hit the *Lookup* button. Publication details will be fetched from PubMed API, resulting in showing PMID and publication title.



2. *Entering the details manually* - click on the *Insert manually* button and enter all the details into the window that appears on the screen.



After setting the publication data, you can modify the data using `View | Show / edit current publication` menu item.

3.2 Genome build

For now, please use build 37 (called either "GRCh37" or "hg19"). Later, we will use the liftover utility of UCSC to add data for build 38.

3.3 Target Gene

In presumably almost all cases, we will know the target gene of the variant that has been published. We enter two bits of information:

- Entrez gene ID (e.g. 3172)
- gene symbol (e.g. *HNF4A*)

Note that the autocompletion is available for both fields, so usually entering just the gene symbol should be enough.

3.4 Variants

Click to *Add variant* button in order to create a new box for variant data. There are several variant types, where we store different set of variant validation metadata for each type.

3.4.1 Mendelian

Validation metadata important for the *Regulatory mendelian mutations (REMM)* project.

3.4.2 Somatic

Validation metadata for somatic variants.

3.4.3 Splicing

Data regarding splicing for the variants curated in the *Squirls* project.

3.4.4 Structural (Intrachromosomal/Translocation)

The way how we store data for variants stored in format denoted as *symbolic* in the VCF specs. These variants are usually longer (>100bp) deletions, duplications, inversions, etc.

We store the variants that affect a single chromosome using `INTRACHROMOSOMAL` variant type. The variants that affect multiple chromosomes (translocations/breakends) are stored as `TRANSCLOCATION` type.

3.4.5 Chromosome and position

Consult the article you are reading. I have found it helpful to see if the sequence surrounding the variant position is shown somewhere in the article. If this sequence is 20 nucleotides or more, you can use the BLAT tool of UCSC Genome Browser to find the corresponding position in the genome. If there are only a few bases, sometimes you can use guesswork to narrow things down enough to find the corresponding place in the genome. For older articles that specify the position of a variant using Genome Build 36 (called either "GRCH36" or "hg18"), you can use the UCSC

Liftover utility. There are some articles that are of such low quality that it is simply not possible to reliably identify the chromosomal position of the variant. In these cases, the article should be rejected. It may also be worthwhile to consult dbSNP or ClinVar, since some published pathogenic variants are entered in these databases.

Note that position should be **one-based**, and *not* zero-based.

3.4.6 Reference / Alternative allele

For single-nucleotide variants, *Ref* and *Alt* are simply A,C,G, or T.

For deletions and insertions, please use the VCF format. Here is the Webpage with the latest details, but if in doubt please ask Peter. Just to give a simple example:

Let us pretend we have a ten base-pair reference sequence on chromosome `Z`:

```
ACGTAAGTCA
```

Let us imagine that the `T` at position 4 is deleted. This results in the sequence:

```
ACGAAGTCA
```

It might seem logical to write simple position=4, ref="T", alt="-". VCF format calls instead for this:

```
#CHROM POS ID REF ALT (other stuff)
Z 3 . GT G (other stuff)
```

This means that the dinucleotide at position 3-4 is affected and the variant sequence has only a G. For an insertion of a C between the T at position 4 and the A at position 5, we write:

```
#CHROM POS ID REF ALT (other stuff)
Z 4 . T TC . (other stuff)
```

We will use this convention, which will allow us to check the reference sequence and the position even for deletions, and should allow us a little more possibilities for Q/C-ing the genomic position etc.

3.4.7 Variant status

We need to enter information about whether the variant is **heterozygous** or **homozygous**. Note that if the patient has two different heterozygous mutations (i.e., is compound heterozygous), then we enter the second mutation in the second *Variant* box. In all other cases, we just use the first *Variant* box. Also, note that in some cases, the publications state (for an autosomal recessive disease) that *"the second mutation could not be found"*. Also in this case, do not enter anything into the second *Variant* box.

Note that if the first mutation is regulatory and the second mutation is coding (e.g., missense, nonsense, splicing, etc.), then you should use the category *coding* for the second mutation.

Finally, it is a good idea to use the Mutalyzer to check the nomenclature and location of the variants. The Mutalyzer will provide the surrounding genomic sequence for most variants, and this can be used to identify the genomic position of coding mutations using BLAT. It may also be useful to consult with ClinVar or the public version of HGMD about this.

3.4.8 Variant class

One of:

1. *promoter* - note that there are no really good definitions of where the promoter is located. Please put anything in the 5UTR in the class 5UTR, even if the effect seems to be on the promoter. Probably anything within 5-10,000 nucleotudes upstream of the transcription start site can be called promoter, but since we will have the numbers, we can do the classification automatically later. For now, I have taken the classification as mentioned in the original publications.
2. *enhancer* - regulatory region that is farther removed from the transcriptional start site than a promoter.
3. *5' UTR*
4. *3' UTR*
5. *microRNAgene* - here we mean any variation that affects the transcript that encodes for a microRNA (note: mutations that affect microRNA binding sites should in general be classified as *3' UTR*).
6. *RNP_RNA* - ribonucleoprotein (RNP) RNA component gene. These include ribosome and snRNP
7. *LINC_RNA* long intergenic non-coding RNA gene
8. *coding* - we only incldue coding mutations if the patient being described was compound heterozygous for a coding mutation and a regulatory mutation

Note that the *5' UTR* DNA sequences often form part of the actual promoter, and in general it is not possible to know if a variant affects the promoter function or the *5' UTR* function (which is of course in the mRNA and can affect the stability of the transcript). If a mutation is located in the *5' UTR*, then please enter *5' UTR* even if the effect is on the promoter. The data base and downstream analysis just has to know about this. In some cases, a mutation may be both *5' UTR* and promoter etc. Please enter the category that seems most relevant. We will automatically generate these annotations using `jannovar` anyway, so even variants with multiple categories will be correctly classified.

Note again that the category *coding* should only be used for the *second* mutation in compound heterozygous cases. At some point we may want to consider adding other classes, but none of the old data will be affected by a new class (e.g., silencer).

3.5 Disease data

Set the database (please use the OMIM id if at all possible). For OMIM, use the phenotype id, and not the gene id.

1. *Database*: one of OMIM or ORPHANET (use drop-down menu)
2. *Disease name*: please use a lower-case form of the canonical name, i.e., do not include all of the synonyms in upper-case letters.
3. *Database ID*: for OMIM; this will be a number like 614321

3.6 Phenotype data (HPO)

To enter or to modify the HPO data, you want and click on the *Add / remove HPO terms* button. Note that if you find you do not have enough, you can add additional terms with this button too.

A new window will be opened with *HPO tree browser* on the left side, *Text-mining analysis* on the right side and with table of *Approved terms* on the bottom-right side.

You should start typing name of the phenotypic trait into the text field above from the ontology tree. The text field has an autocompletion feature and helps you to identify the correct *HPO term label*. After completion of the label, click on the *Go* button to navigate to the term's position in the ontology tree.

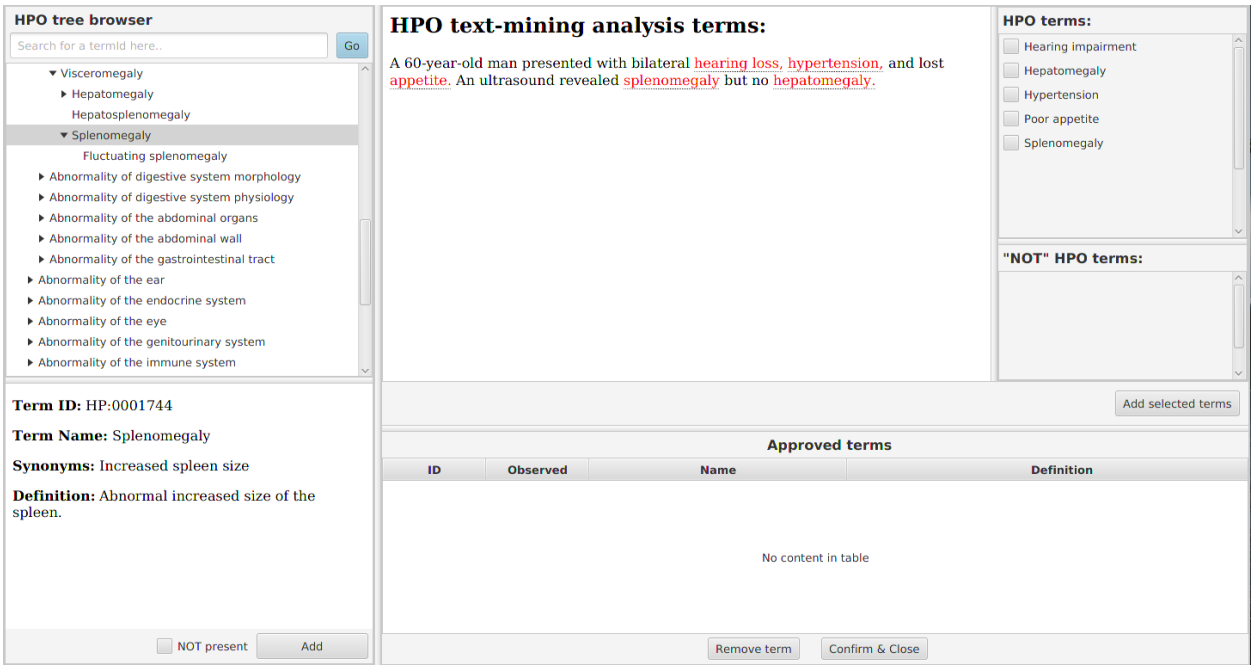
Then, you may want to look around the term in the ontology tree a bit and then approve the term's presence by hitting *Add* button at the bottom. The term will appear in the *Approved terms* table.

3.6.1 Text mining

In case you're curating variants from a publication that contains a clinical description of the proband's condition, *text mining* comes to help. To identify candidate HPO terms in a clinical description text, paste the text into the *Text-mining analysis* field.

Try the text-mining using e.g. the following toy example:

```
A 60-year-old man presented with bilateral hearing loss, hypertension, and lost␣
↪appetite.
An ultrasound revealed splenomegaly but no hepatomegaly.
```



Five HPO terms are picked up from the toy example. HPO term definition appears upon hovering with mouse upon the highlighted text. Clicking on the text will navigate you to the term definition within the ontology hierarchy (left panel). We recommend to read the text, approve the relevant terms on the right panel, and approving the mined terms by clicking on *Add selected terms* button.

Note: The previously used text-mining service was also able to identify *not* terms (e.g. no hepatomegaly). Unfortunately, the current service does not support this feature.

3.7 Proband & Family Information

The ID (patient/family identifier) is a free-text string that represents the ID used to designate the affected individual or family in the original paper. For instance, `family 3`. Note that we usually include all of the pathogenic variant in a given paper, but if little clinical data is given, and the phenotype is identical for two families, then it is OK to enter `family 3 and family 7`, say.

3.8 Metadata

Many of the individual papers about disease-causing variants have a lot of interesting additional information that is more or less heterogeneous. We would like to capture the most salient points in a free text that will be displayed on the planned website. For instance, here is an example `Metadata`:

```
The mutation is located in a 400-bp sequence located 25 kb downstream of PTF1A (the␣
↪gene
for pancreas-specific transcription factor 1a). This region acts as a developmental␣
↪enhancer
of PTF1A and that the mutations abolish enhancer activity. The mutation was shown to␣
↪abolish
binding of FOXA2 (Supplementary Figure 8 of Wheedon et al., 2014).
```

CHAPTER 4

Validation

Warning: The documentation has not been updated for the *2.** version yet.

For some of the uses of `HpoCaseAnnotator`, we enter not only the phenotype and genotype information, but also information about the molecular pathomechanism of the variant as well as any experimental methods that were used to validate the pathogenicity of the variant.

4.1 Non-coding variants

We have curated many non-coding variants that were used to validate the Genomiser. As a rule, we only include a mutation if there is adequate evidence for its pathogenicity. As a general rule, there should be some experimental evidence for the mutation changing gene regulation of a target gene in some way. For some heavily studied genes, we will accept a mutation if it seems to be very similar to other published mutations (e.g., it lies on the same predicted transcription factor binding site as another mutation for which experimental evidence is available). Add as much evidence as possible. It is expected that at least one of the evidence categories will apply to each mutation.

1. reporter - Luciferase assay (or the similar CAT assay) to judge transcriptional activity. Indicate whether the mutation is associated with increased activity (up) or decreased activity (down) as compared with the wildtype construct (in percent).
2. EMSA - EMSA (electrophoretic mobility shift assay). This is used to indicate whether a protein binds to a given DNA sequence. For our purposes, we are referring to the protein affected by the mutation. Enter the corresponding protein if there is a change in binding. Enter the Entrez Gene ID and Gene Symbol of the protein that is affected by the mutation (usually a transcription factor)
3. cosegregation - enter yes if the mutation cosegregates with the disease in the family being investigated.
4. comparability - this is the weakest evidence class. Enter yes if the reason for believing that the mutation is pathogenic is simply that it is comparable to other published regulatory mutations in the gene.
5. other - this is for any other kind of experimental assay that shows an effect of a regulatory or non-coding mutation. Note that for now the categories are hard coded into the Java code, this should be put into some kind

of configuration file in the future. The categories are at present:

Telomerase. Telomerase lengthening assay.

4.2 Splicing variants

For splicing variants, we include them if there is adequate evidence for missplicing and disease pathogenicity. TODO – describe.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search