VariantGrid Admin Docs

CCB ACRF Cancer Genomics Facility

Oct 03, 2023

ADMIN:

This documentation is for admins who configure a VariantGrid server.

For normal users, see VariantGrid user documentation or Developers Technical Wiki

CHAPTER

ONE

MANAGING USERS

All users are automatically added to the "public" group.

1.1 Built in Authentication

For normal users, staff/admin users can create further users via the admin interface. This is "/admin" after the URL, eg: https://variantgrid.com/admin/ (need to change to your server obviously)

In the "Authentication and Authorization" section, there's a line with "Users" - click the "+ Add" link, then enter the user/details. Groups

```
You may want to add people to different groups so they can share data between themselves.

Users, permissions and groups

User / Lab Assignment
```

1.2 Windows Authentication (LDAP)

LDAP (Lightweight Directory Access Protocol) is the protocol behind Microsoft Active Directory, which is what most corporate intranets use for logins, Outlook, shared drives etc.

-

See Wiki: LDAP Setup

1.3 KeyCloak Authentication

CHAPTER

TWO

SERVER STATUS

CHAPTER

THREE

CHAPTER

FOUR

DJANGO ADMIN (ADMIN ONLY PAGE)

Django comes with an admin interface that allows users (with is_staff=True) to modify database records.

Programmers need to register the models, so let them know if something doesn't appear.

The interface is automatically generated, but the look and behavior can be customised (eg to use an autocomplete field instead of a select)

4.1 Admin Actions

Generally, you use Django admin to edit a single record at a time, but you can also create actions and apply them to multiple records (by selecting them) or all records.

VariantGrid actions are documented in individual sections, eg the "reattempt_variant_matching" action is documented in *Classification Admin Pages*.

CHAPTER

FIVE

SCANNING AND AUTO-LOADING SEQUENCING FILES

5.1 Process overview

5.2 Scanning filesystem

What gets scanned

How frequently. Manually

5.3 Troubleshooting

Errors

Deleting and restarting

How do we reload something

CHAPTER

SIX

SAMPLESHEET.CSV

Illumina SequencingRuns have a file called "SampleSheet.csv" which lists the

Many years ago we noticed this file became the main handover point between wet and dry lab staff, and asked Illumina if they would ignore any extra columns in this file, so we can add as many extra columns as we want to annotate our samples.

6.1 Special Columns

Enrichment Kits are assigned to rows in the SampleSheet.csv via the *Panel* column. If all samples are assigned to the same kit, the SequencingRun is also assigned to that kit.

Some panels can change over time (eg it's a custom kit or the manufacturer adds some new genes) - in which case you can use panel versions.

To do this, add a **PanelVersion** field in the SampleSheet.csv

An EnrichmentKit is a lab method to enrich a sample for the DNA regions you are interested in. For instance an exome or custom gene capture kit, or amplicons.

CHAPTER

SEVEN

ENRICHMENTKIT SETUP

EnrichmentKits are automatically created based on directories flowcells are detected in via scanning sequencing dirs, and entries in the sample_sheet

You need to upload:

- A Bed containing capture regions
- A GeneList containing symbols captured

If the canonical transcripts file contains only genes in the kit, you can turn it into a gene list via:

```
grep -v "^#" enrichment_kit.GeneTable.tsv | cut -f 1 > /tmp/gene_list.txt
```

Setup canonical transcripts (if this differs from existing canonical transcripts)

```
python3.8 manage.py import_canonical_transcript --genome-build=GRCh38 --annotation-
↪consortium=RefSeq enrichment_kit.GeneTable.tsv
```

In Django admin, go to **seqauto** -> **EnrichmentKits** then select the kit.

Select the appropriate details for your kit (eg bed/genelist etc you uploaded) and click save.

CHAPTER

EIGHT

REFSEQ VS ENSEMBL

CHAPTER

NINE

ANNOTATION VERSIONS

What happens when you upgrade annotation versions?

Keep the old one? How much space it uses etc

Might want to delete etc

CHAPTER

TEN

ANNOTATION VERSIONS

What happens when you upgrade annotation versions?

Keep the old one? How much space it uses etc

Might want to delete etc

CHAPTER

ELEVEN

GENE COVERAGE

CHAPTER

TWELVE

GENE LISTS

CHAPTER

THIRTEEN

PATIENT RECORD SYSTEM INTEGRATION

CHAPTER

FOURTEEN

ANALYSIS - ISSUES (ADMIN ONLY PAGE)

Admin only page - shows node failures

Reload them - what does it mean

CHAPTER

FIFTEEN

ANALYSIS - TEMPLATES

Any user can create an analysis template, but only administrators can make them “official” and appear as links (on the sample/Cohort page etc)

This is done at an Template Version level - edit via admin at eg https://variantgrid.com/admin/analysis/analysistemplateversion/

You can alter the following check boxes:

- Appears in autocomplete
- Appears in links
- Requires sample somatic
- Requires sample gene list

CHAPTER

SIXTEEN

CLASSIFICATION ADMIN PAGES

16.1 Activity

This page shows changes that have been performed on a classification (or classifications) evidence and flags. The oldest changes are shown at the top and the newest at the bottom.

16.2 HGVS Issues

This page shows Alleles with open flags that require human action/validation.

A typical flag is “Mismatched c.HGVS” (37/38) - when you liftover between genome builds at the coordinate level, sometimes the c.HGVS coordinate can change (due to mRNA transcripts aligning differently to the reference sequences, causing eg exon boundary changes). We flag this for a human to double check everything is ok.

16.3 Django Admin tasks

CHAPTER

SEVENTEEN

CLASSIFICATION REPORT

17.1 Configuring the report

The report can only be configured by admin users. Each "organisation" within variantgrid uses its own report. To edit it go to the admin view, Organisations, (your organisation), and then edit the Classification report template.

The template is run using Django template and produces HTML

17.2 Values available for the report

17.2.1 Evidence Keys

All the fields in the classification are exposed here, see the Evidence Keys admin for a list of possible values, e.g. zygosity, mechanism_of_disease, mode_of_inheritance. In addition you can also suffix `_raw` or `_note` e.g.

```
The raw value for Mode of Inheritance is {{ mode_of_inheritance_raw }} and the note for␣
↪it is {{ mode_of_inheritance_note }}
{% if mode_of_inheritance_raw == 'x_linked' %}
Special case for X Linked
{% endif %}
```

Typically you'll only want to refer to the `_raw` value if you're doing some logic for a specific drop down value. If you ommit the `_raw` then you will get the human friendly label for the value which might subtly change in the future.

17.2.2 p.hgvs

You can reference the full `p_hgvs` or breakdown

```
full p.hgvs = {{ p_hgvs }}<br/>
p amino acid from = {{ p_hgvs_aa_from }}<br/>
p hgvs codon = {{ p_hgvs_codon }}<br/>
p hgvs amino acid to = {{ p_hgvs_aa_to }}
```

17.2.3 c.hgvs

You can reference the full `c_hgvs` or breakdown

```
full c.hgvs = {{ c_hgvs }}<br/>
c hgvs transcript = {{ c_hgvs_transcript }} or {{refseq_transcript_id}}<br/>
c hgvs gene symbole = {{ c_hgvs_gene_symbol }} or {{ gene_symbole }}<br/>
c hgvs short = c.{{ c_hgvs_short }} (this is the value in c_hgvs after "c.")
```

17.2.4 Evidence weights

A summary of the strength of ACMG critieria met can be accessed with

```
Evidence weights = {{ evidence_weights }}
```

17.2.5 Citations

PMIDs put anywhere in the classification can be accessed, and then specific attributes of those citations can be referenced. `citations` is an array that you must loop through, e.g.

```
{% for cit in citations %}
        <tr>
                <td>{{ cit.source }}</td>
                <td>{{ cit.citation_id }}</td>
                <td>{{ cit.citation_link }}</td>
                <td>{{ cit.journal }}</td>
                <td>{{ cit.journal_short }}</td>
                <td>{{ cit.title }}</td>
                <td>{{ cit.year }}</td>
                <td>{{ cit.authors }}</td>
                <td>{{ cit.authors_short }}</td>
                <td>{{ cit.abstract }}</td>
        </tr>
{% endfor %}
```

The example here is in a table but you can display it however you'd like, e.g.

```
{% for cit in citations %}
{{ cit.source }}:{{ cit.citation_id }}
{% endfor %}
```

Which would give you PMID:12334 PMID:4555 etc

CHAPTER

EIGHTEEN

INDICES AND TABLES

- genindex
- modindex
- search