PQuery

PQuery is a Flask-based web application enabling fast interactive visualisation, querying, flexible filtering and alignment inspection of multi-sample, annotated, genetic variant data.

Without programming skills required, users can automatically convert their input VCF file (compressed or uncompressed, annotated or not annotated) to an indexed and structured SQLite database, fully adjusted to the imported VCF file. Once the database is created and stored, PQuery can repeatedly open in a browser window and instantly run on it with no need for re-importing every time it runs. The application allows users to query the genetic variants for specific samples and genomic regions and it then instantly returns a table with the queried variants (one variant per row) and all fields present in the original VCF file. It also dynamically calculates the allele counts and frequencies for the selected samples.

PQuery can run in two modes:

a. The 'cohort mode' in which no sample-specific information is included in the returned table
b. The 'sample-specific' mode in which the generated table will fully include the sample names and all the related information of the original VCF file.

Users can efficiently filter all columns in both modes by using a customisable filter pane and inspect the variant calls through the integrated IGV viewer.

PQuery

Getting Started

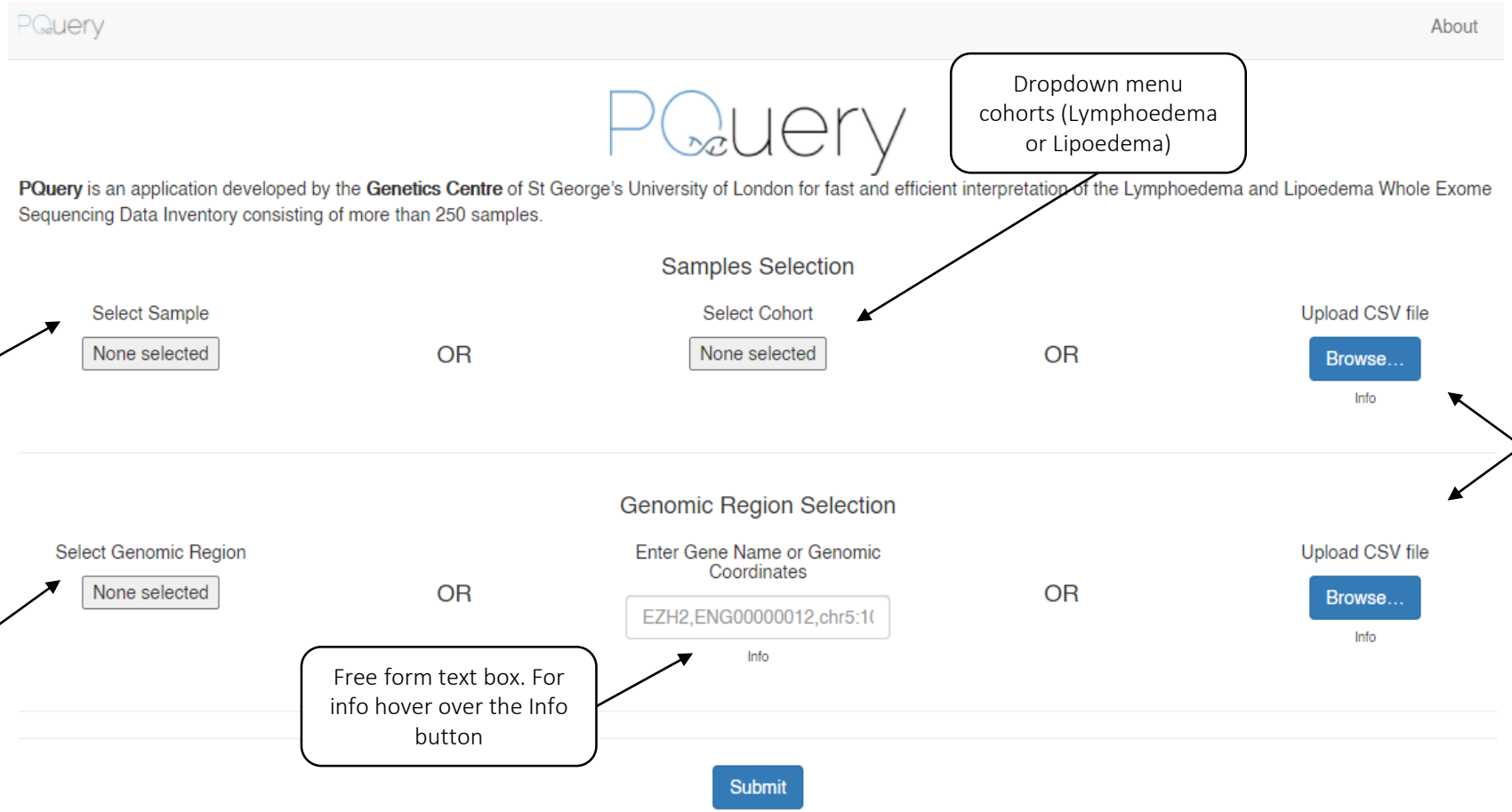
How Execute PQuery

- From a linux machine: `sh run_app.sh`
- From a Windows machine: Double-click the PQuery.bat file or the PQuery.lnk shortcup (you can copy the shortcut to your Desktop).

PQuery

- **PQuery** is a flask web application (prototype) built to visualise annotated multi-sample variant-level data stored in VCF files.
- It's really fast for querying small regions like multiple genes.
- User can select genomic regions and samples to query:
  - If <=40 samples selected, PQuery will run in **sample mode** and the output table will include per-sample genotype information.
  - If >40 samples selected, PQuery will run in **cohort mode** and the output table will not include per sample genotype columns in the output table.
- It is better to query small regions rather than the whole exome which will be slow.
- PQuery is querying the hg19 POlab data.

Home Page



Dropdown menu cohorts (Lymphoedema or Lipoedema)

Dropdown menu with all samples (with embedded search box)

Update CSV file with sample names or genomic locations (one per line) – Hover over the info buttons on each field for more info.

Dropdown menu all chromosomes and the option to query "Whole Exome"

Free form text box. For info hover over the Info button

- You have to use **only one** of the proposed ways to enter your input for samples and/or for genomic regions.
- You need to specify at least one sample and at least one valid genomic regions or the app will throw an error.
- Hit submit when you enter your input.
- Everytime you hit submit (or refresh) the page is being refresh and the input is being cleared.

This is the var_table page with the query result: An unltra-wide annotated variants table

Download the table as CSV button.

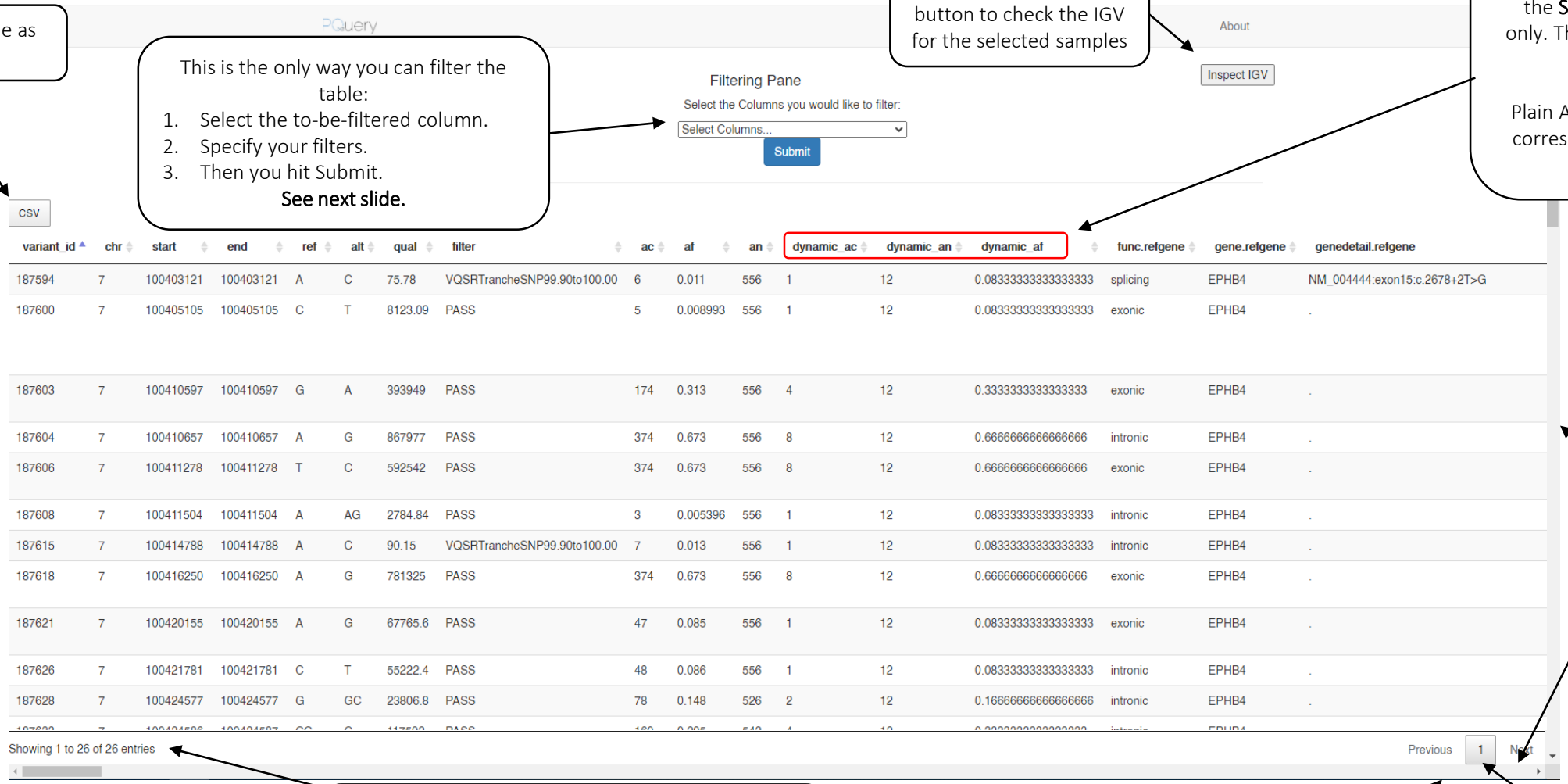
This is the only way you can filter the table:
1. Select the to-be-filtered column.
2. Specify your filters.
3. Then you hit Submit.

See next slide.

Click the Inspect IGV button to check the IGV for the selected samples

Columns showing the Alt. Allele Count (**AC**), Total Allele Number (**AN**) and Allele Frequency (**AF**) for the **SELECTED samples** only. They are **dynamically** calculated.

Plain AC, AF, AN columns correspond to the whole cohort.



variant_id	chr	start	end	ref	alt	qual	filter	ac	af	an	dynamic_ac	dynamic_an	dynamic_af	func.refgene	gene.refgene	genedetail.refgene
187594	7	100403121	100403121	A	C	75.78	VQSRTrancheSNP99.90to100.00	6	0.011	556	1	12	0.08333333333333333	splicing	EPHB4	NM_004444:exon15:c.2678+2T>G
187600	7	100405105	100405105	C	T	8123.09	PASS	5	0.008993	556	1	12	0.08333333333333333	exonic	EPHB4	.
187603	7	100410597	100410597	G	A	393949	PASS	174	0.313	556	4	12	0.3333333333333333	exonic	EPHB4	.
187604	7	100410657	100410657	A	G	867977	PASS	374	0.673	556	8	12	0.6666666666666666	intronic	EPHB4	.
187606	7	100411278	100411278	T	C	592542	PASS	374	0.673	556	8	12	0.6666666666666666	exonic	EPHB4	.
187608	7	100411504	100411504	A	AG	2784.84	PASS	3	0.005396	556	1	12	0.08333333333333333	intronic	EPHB4	.
187615	7	100414788	100414788	A	C	90.15	VQSRTrancheSNP99.90to100.00	7	0.013	556	1	12	0.08333333333333333	intronic	EPHB4	.
187618	7	100416250	100416250	A	G	781325	PASS	374	0.673	556	8	12	0.6666666666666666	exonic	EPHB4	.
187621	7	100420155	100420155	A	G	67765.6	PASS	47	0.085	556	1	12	0.08333333333333333	exonic	EPHB4	.
187626	7	100421781	100421781	C	T	55222.4	PASS	48	0.086	556	1	12	0.08333333333333333	intronic	EPHB4	.
187628	7	100424577	100424577	G	GC	23806.8	PASS	78	0.148	526	2	12	0.16666666666666666	intronic	EPHB4	.

Showing 1 to 26 of 26 entries

Previous 1 Next

Use the normal Scrollbars to navigate through the table in both dimensions.

Use your browser's find functionality (Ctrl+F) to search for column names

Entries counter: BUG
This is **not** showing the right number of variants returned.
It shows the number of variants BEFORE they are filtered for AC>0. Ignore for now.

Scroll to the very left end of the table to see the **genotypes per variant of the selected samples IF RUNNING ON THE SAMPLES mode.**

Paging options

PQuery

About

Inspect IGV

Filtering Pane

Select the Columns you would like to filter:

D102715

ref

func.refgene: ncRNA_exonic, ncRNA_intronic, exonic, intergenic

gene.refgene

cadd_phred From: To:

D102715 Select...

Submit

There are 4 types of filter predefined for each column:

1. Free form text (For example for the REF/ALT columns).
2. Multi-select dropdown menu.
3. Numeric Ranges (From, to) If you want you can only specify 'from' that equals to '>=' or 'to' that equals to '<=' leaving the other one empty.
4. Sample Genotypes filter.

Click **Submit** to apply the filters. Whenever you click submit all the specified filtering is being cleared and YOU NEED TO RESPECIFY all of them if you want to add one more filter for example.

Use these small arrows to **sort** the columns. You can do that on **every column** on the table

CSV

variant_id	chr	start	end	ref	alt	qual	filter	ac	af	an	dynamic_ac	dynamic_an	dynamic_af	func.refgene		
187603	7	100410597	100410597	G	A	393949	PASS	174	0.313	556	3	10	0.3	exonic		
187604	7	100410657	100410657	A	G	867977	PASS	374	0.673	556	7	10	0.7	intronic		
187606	7	100411278	100411278	T	C	592542	PASS	374	0.673	556	7	10	0.7	exonic		
187615	7	100414788	100414788	A	C	90.15	VQSRTrancheSNP99.90to100.00	7	0.013	556	1				EPHB4	.
187618	7	100416250	100416250	A	G	781325	PASS	374	0.673	556	7				EPHB4	.
187621	7	100420155	100420155	A	G	67765.6	PASS	47	0.085	556	1	10	0.1	exonic	EPHB4	.
187626	7	100421781	100421781	C	T	55222.4	PASS	48	0.086	556	1	10	0.1	intronic	EPHB4	.

PQuery

About

Select the Samples you would like to check IGV for:

None selected

And then click Submit:

Submit

Click here to show the list of samples which can be selected for IGV.

Select the samples.

Then click Submit.

After clicking submit you can set-up IGV. For more information on how to use IGV WebApp check here: https://igvteam.github.io/igv-webapp/