Spec for query region -by alias

1 Definition

This query will use aliases to refer to genomic regions coordinates that will be mapped to a `query by region -(by coordinates)`, so it can be considered a special case of `query by region` and be implemented on top of it. The mapping of names/aliases to `query by region` parameters (**refseqId** and **start**, **end** positions) comes from genome annotation file.

Note: This query type implementation could be deprecated after we have better annotated VCFs that include mature proteins and possibly other functional domains. Then, `query by annotation`, an implementation that uses variant annotations in VCF regardless of RefSeqID, can substitute for this.

2 Query specifications

Parameters: same as in `query by region`, i.e **assemblyId** and/or **refseqId**, **start** and **end**.

For the time being **refseqId** is set to default value since we only have one virus (taxonId:2697049) with just one RefSeq (refseqId:NC_045512.2) in Viral Beacon. Note: When/if more viruses come (and also for generic Beacon) user will have to enter those parameters as well (so there should be a table in backend mapping all available **taxonId** and their available **assemblyId** and other table mapping assemblyIds to their corresponding sets of **refseqId**).

User input: **assemblyId** or **refseqId** (default now) and an alias substituting for **start** and **end** in `query by region`.

Frontend suggestion: User enters a genomic region name/alias/identifier/accession in query box like in figure 1. Either a dropdown menu or a suggesting-while-typing menu should appear showing the query 'options' based on the mapping table stored in backend for this **refseqId**.

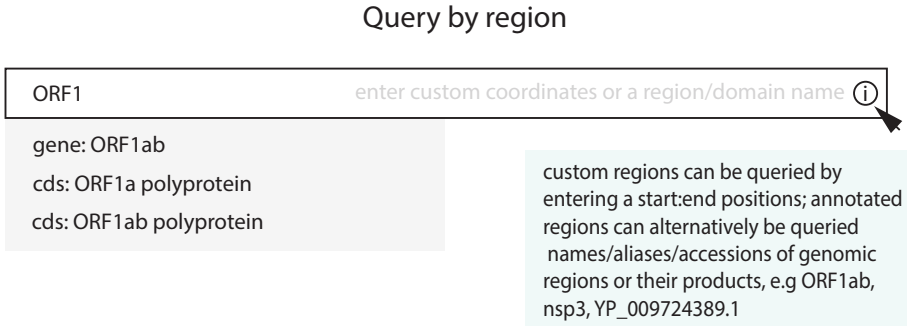

Figure 1: Query by region box

(Later, when we have implementation for multiple `query by region` at once -as for `query by motif`- the idea would be being able to allow search also for lists of those aliases (e.g entered separated by commas), and for grouping terms such as "intergenic", "coding", "non-coding", "utr", "stem loops" or "structural protein", that would translate into a list of start:end coordinates)

3 Prerequisites

1 A table/dictionary mapping names/aliases or synonyms/accessions accepted in this field to genomic region coordinates.

To do this, the genomic annotation file for the specified **refseqId** was fetched. For SARS-CoV2 we will use reference sequence NCBI RefSeq NC_045512.2 annotation available at NCBI web. This annotation contains the **refseqId** ("NC_045512.2") and the names/ aliases of genomic regions with their corresponding **start** and **end** coordinates.
The relevant data was parsed as a table like in table 1 (download the table as csv).

4 Implementation

1 Entries from table should be converted to 'options' for frontend. The 'option' would work as a dictionary key for all names/aliases in the table row. The 'option' is what should appear as suggestions while user types a part of either `name`, `syn_alias`, `locus_tag`, `id` or `accession` of its row, so some flexibility is allowed. Alternatively, 'options' will appear in a dropdown menu (so, no flexibility but user knows his options).

'Options' will be constructed by concatenating 'value in `class`:value in `name`' from table 1, eg. gene: ORF8, cds: ORF1a polyprotein, functional: RNA-dependent RNA polymerase, functional: Coronavirus 3' UTR pseudoknot stem-loop 1, non-coding: 5'UTR.

2 Coordinates (`start` and `end`) will be pulled from table 1 using the 'option' `name` and then `query by region -(by coordinates)` is run as usual using these as parameters `start` and `end`.

5 Response

Response will be same as that of a `query by region`.

Table 1: Genomic regions coordinates and aliases

class	type	start	end	name	
gene	coding	266	21555	ORF1ab	
gene	coding	21563	25384	S	spike gly
gene	coding	25393	26220	ORF3a	
gene	coding	26245	26472	E	
gene	coding	26523	27191	M	
gene	coding	27202	27387	ORF6	
gene	coding	27394	27759	ORF7a	
gene	coding	27756	27887	ORF7b	
gene	coding	27894	28259	ORF8	
gene	coding	28274	29533	N	
gene	coding	29558	29674	ORF10	
cds	product	266	21555	ORF1ab polyprotein	pp1ab
cds	product	266	13483	ORF1a polyprotein	pp1a
functional	mature peptide	206	805	leader protein	leader, n
functional	mature peptide	806	2719	nsp2	nsp2
functional	mature peptide	2720	8554	nsp3	nsp3
functional	mature peptide	8555	10054	nsp4	nsp4
functional	mature peptide	10055	10972	3C-like proteinase	nsp5, ma
functional	mature peptide	10973	11842	nsp6	nsp6
functional	mature peptide	11843	12091	nsp7	nsp7
functional	mature peptide	12092	12685	nsp8	nsp8
functional	mature peptide	12686	13024	nsp9	ssRNA-b
functional	mature peptide	13025	13441	nsp10	nsp10
functional	mature peptide	13442	13480	nsp11	nsp11
functional	mature peptide	13442	16236	RNA-dependent RNA polymerase	RdRp
functional	mature peptide	16237	18039	helicase	nsp13, h
functional	mature peptide	18040	19620	3'-to-5' exonuclease	3'-5' exo
functional	mature peptide	19621	20658	endoRNAse	nsp15
functional	mature peptide	20659	21552	2'-O-ribose methyltransferase	2'-o-MT,
cds	product	21563	25384	surface glycoprotein	Spike
cds	product	25393	26220	ORF3a protein	
cds	product	26245	26472	envelop protein	E protein
cds	product	26523	27191	membrane glycoprotein	membran
cds	product	27202	27387	ORF6 protein	
cds	product	27394	27759	ORF7a protein	
cds	product	27756	27887	ORF7b protein	
cds	product	27894	28259	ORF8 protein	
cds	product	28274	29533	nucleocapsid phosphoprotein	nucleoca
cds	product	29558	29674	ORF10 protein	
non-coding	utr	1	265	5'UTR	
non-coding	utr	29675	29903	3'UTR	
functional	stem loop	13476	13503	Coronavirus frameshifting stimulation element stem-loop 1	fsSE SL1
functional	stem loop	13488	13542	Coronavirus frameshifting stimulation element stem-loop 2	fsSE SL2
functional	stem loop	29609	29644	Coronavirus 3' UTR pseudoknot stem-loop 1	3utr pk S
functional	stem loop	29629	29657	Coronavirus 3' UTR pseudoknot stem-loop 2	3utr pk S
functional	stem loop	29728	29768	Coronavirus 3' stem-loop II-like motif (s2m)	s2m