Portal/Hail API

Version	Date
0.1	2018-04-17

Retrieve available datasets/masks

A dataset is a set of variant genotypes and samples (e.g. a VCF.)

A mask is defined by:

1. Variant filtering criteria
2. Variant grouping criteria

Variant filtering:

- Allele frequency
- Annotation about a variant (protein-truncating, loss-of-function, etc.)

Grouping:

- Gene
- Chromatin state regions (enhancer, silencer, promoter, insulator, etc.)
- KEGG pathways (variants within genes that belong to the pathway)
- Gene ontology terms (similar to KEGG)

Masks would be available per dataset.

For now, a description of the variant filtering and grouping criteria will probably just have to be plain text without some type of formal grammar.

Request

```
GET /api/aggregation/metadata
```

Response


```
{
  "data": [
    {
      "dataset": 42,
      "analysis": "52K Exomes",
      "masks": [
        {
          "id": "PTV",
          "description": "Protein truncating variants",
          "grouping": "gene",
          "identifier": "ENSEMBL"
        },
        {
          "id": "PTV & LoF & AF<0.05",
          "description": "Protein truncating variants with AF < 0.05",
          "grouping": "gene",
          "identifier": "ENSEMBL"
```

```
        },
        {
          "id": "PTV & LoF & AF<0.05",
          "description": "Protein truncating and loss-of-function variants with AF < 0.05",
          "grouping": "gene",
          "identifier": "ENSEMBL"
        }
      ]
    }
  ]
}
```

Retrieve covariance in region given dataset/mask(s)

Request

Only requesting scores and covariance for the "PTV" mask just to save some space.

`POST /api/aggregation/covariance`

```
{
  "chrom": "6",
  "start": 1,
  "end": 100000,
  "dataset": 42,
  "masks": [
    "PTV"
  ]
}
```

Response

We want to be able to retrieve as much of the masks/covariance data all at once to minimize round trips to the server. This structure would allow retrieving covariance for multiple masks all at once.

Note that all scores and covariances are assumed to be counting towards the **alternate allele**. It is also critical that alternate allele frequencies are included so as to be able to orient scores/covariances towards the minor allele when performing the aggregation tests.

This response is slightly condensed to save space.

```
{
  "data": {
    "masks": [
      {
        "id": "PTV",
        "label": "Includes only protein truncating variants",
        "groups": {
          "ENSG000001": {
            "variants": [
              "2:21228642_G/A",
              "2:21230094_AT/A",
              "2:21230336_AT/A"
            ]
          },
```

```
      "ENSG000002": {
        "variants": [
          "19:11216264_G/T",
          "19:11218173_AACCCATC/A"
        ]
      }
    }
  }
],
"scorecov": [
  {
    "mask": "PTV",
    "group": "ENSG000001",
    "scores": [
      0.1,
      0.3,
      0.5
    ],
    "covariance": [
      0.3,
      0.4,
      0.65,
      0.1,
      0.83,
      0.99
    ],
    "altFreq": [
      0.033,
      0.001,
      0.00045
    ],
    "sigmaSquared": 0.08,
    "nsamples": 3550
  },
  {
    "mask": "PTV",
    "group": "ENSG000002",
    "scores": [
      0.443,
      0.911
    ],
    "covariance": [
      0.291,
      0.384,
      0.14
    ],
    "altFreq": [
      0.0025,
      0.03
    ],
    "sigmaSquared": 0.08,
    "nsamples": 3550
  }
]
```

```
  }
}
```

Retrieving pre-computed aggregation test results

In the case where aggregation tests have already been pre-computed for some of the datasets server-side, we could use this API to retrieve those results and display them in LZ.

Note that raremetal.js, after running aggregation tests, will return results formatted in the exact same format as the response below.

Request

Looks identical to the same request used to retrieve covariance.

`POST /api/aggregation/results`

```
{
  "chrom": "6",
  "start": 1,
  "end": 100000,
  "dataset": 42,
  "masks": [
    "PTV"
  ]
}
```

Response

If results were already available server-side, say if they were pre-computed, we could accept a response of this format and be able to show the results within LZ without performing any aggregation test computations.

This response could contain a little more annotation for the individual variants (for example, allele frequencies below.)

```
{
  "id": 42,
  "description": "52K Exomes",
  "data": {
    "masks": [
      {
        "id": "PTV",
        "label": "Includes only protein truncating variants",
        "groups": {
          "ENSG000001": {
            "variants": [
              "2:21228642_G/A",
              "2:21230094_AT/A",
              "2:21230336_AT/A"
            ],
            "altFreq": [
              0.033,
              0.001,
              0.00045
```

```
            ]
          },
          "ENSG000002": {
            "variants": [
              "19:11216264_G/T",
              "19:11218173_AACCCATC/A"
            ],
            "altFreq": [
              0.0025,
              0.03
            ]
          }
        }
      }
    ],
    "results": [
      {
        "group": "ENSG000001",
        "type": "gene",
        "mask": "PTV",
        "test": "SKAT-O",
        "pvalue": 1.8e-09
      },
      {
        "group": "ENSG000002",
        "type": "gene",
        "mask": "PTV",
        "test": "SKAT-O",
        "pvalue": 1.7883e-09
      }
    ]
  }
}
```