GENOMENON®

Mastermind Integration

Technical Documentation

August 23, 2019

hello@genomenon.com | www.genomenon.com | 1-734-794-3075

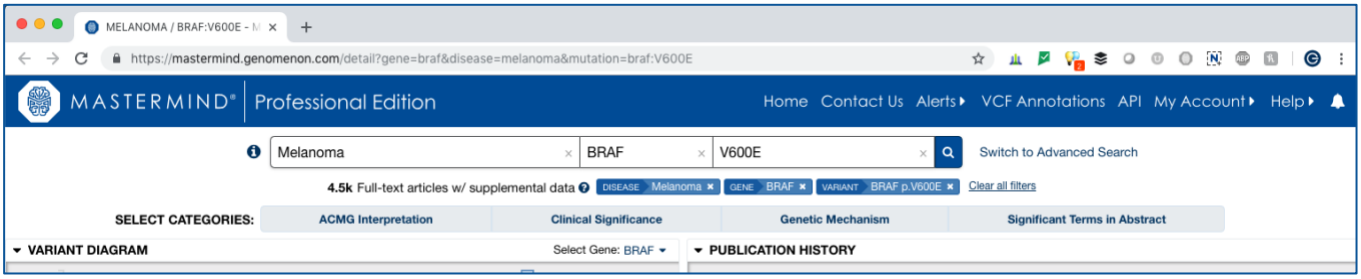
GENOMENON®

Linking to Mastermind

Link Structure

The Mastermind Genomic Search Engine provides structured URLs which allow easy deep-linking into search results for a given disease, gene, and/or variant. For example, here is a URL which takes the user directly to articles for Melanoma associated with the BRAF:V600E variant:

https://mastermind.genomenon.com/detail?disease=melanoma&gene=braf&mutation=braf:V600E



This URL format can be used to dynamically generate deep-links into Mastermind searches by simply replacing the disease, gene, and variant in the URL.

https://mastermind.genomenon.com/detail?disease={disease_name}&gene={gene_symbol}&mutation={gene_symbol:variant_key}

Any of the parameters may be omitted, with the following constraints:

- For Free Edition users, the disease must be specified as "all%20diseases" (which is the URL-encoding of "all diseases") *and* the gene must be specified. The variant parameter may be omitted.
- For Professional Edition users, either one of at least the disease *or* gene must be specified. The other parameter may be omitted for a list of genes associated with the specified disease, or diseases associated with the specified gene.
- For variant searches, the gene parameter must also be specified for the variant.

Referral Code

When integrating Mastermind links into your application, add the `ref` parameter value to the URL. We will set up a unique code for your organization within Mastermind:

https://mastermind.genomenon.com/detail?disease=melanoma&gene=braf&mutation=braf:V600E**&ref=your-org-code**

Parameter Values and Formats

The disease, gene, and variant parameters should follow these rules:

Disease

The value for the disease should be the canonical MeSH term for the disease, in lower-case and URL-encoded (e.g. spaces are encoded as "%20").

To view results irrespective of citations of any specific disease, the value of "all%20diseases" may also be encoded. This will show results for all diseases together.

For help with any specific disease, gene, or variant, try typing the desired value into the appropriate field in the homepage search interface to see which valid values are suggested as part of the auto-complete drop-down.

Gene

The value for the gene parameter should be the desired HGNC gene symbol in lowercase.

Variant

The value for the variant parameter should include the same gene symbol from the The value for the variant parameter should include the same gene symbol from the gene parameter followed by a colon, followed by the variant key. The variant key may be specified in any of the following nomenclatures:

- Mastermind gene variant nomenclature, which is based on HGVS protein (effect) level nomenclature and optimized for search sensitivity.

  For most variants, this is simply the protein nomenclature. However, to maximize sensitivity for specific types of deletions, insertions, duplications, frame-shifts, and non-coding variants, there is a simple transformation to Mastermind variant nomenclature, outlined in the FAQ.
  Example:
  https://mastermind.genomenon.com/detail?disease=all%20diseases&gene=braf&**mutation=braf:E46int**

- HGVS protein "p." nomenclature.
  Example:
  https://mastermind.genomenon.com/detail?disease=all%20diseases&gene=cftr&**mutation=cftr:p.Lys684dup**

- HGVS cDNA "c." nomenclature.
  Example:
  https://mastermind.genomenon.com/detail?disease=all%20diseases&gene=brca1&**mutation=brca1:c.5382insC**

- Reference SNP cluster ID "rsID" identifier from dbSNP.
  Examples:
  https://mastermind.genomenon.com/detail?disease=all%20diseases&gene=cdkn2a&**mutation=cdkn2a:rs3731249**

  https://mastermind.genomenon.com/detail?disease=all%20diseases&**mutation=rs3731249**

- Genomic coordinates using GRCh37. The sequence identifier should be used for genomic coordinates. This is NC_00012.11 for chromosome 12 below:
  Example (the `%3E` below is the URL-encoding of the `>` character):
  https://mastermind.genomenon.com/detail?disease=all%20diseases&**mutation=NC_000012.11:g.57489193T%3EC**

Advanced Search

In addition to the standard disease/gene/variant searches described above, Mastermind Professional Edition also offers advanced search functionality for:

- Searching multiple diseases, genes, and/or variants;
- ACMG/AMP guideline-based searching;
- Filters for clinical significance and genetic mechanisms, such as amplifications, CNVs, and more;
- Exporting of PMID lists with supporting information for any search.

All of these search features are able to be used, combined, and patterned directly within the URL of the links into Mastermind, provided that the users clicking the links have access to Mastermind Professional Edition. If the user has access only to Mastermind Free Edition, upon clicking, they will be shown a message that the linked search requires Professional Edition to view.

Single Sign-on

Benefits

Single Sign-on (SSO) integration into Mastermind allows users of your platform to move seamlessly back and forth between your platform and Mastermind without needing to sign up for or log into Mastermind separately. Their account in your platform acts as their account in Mastermind.

This works by linking to or embedding Mastermind from within your application using special Mastermind SSO URLs.

Getting Started

1. Contact us to configure SSO access to Mastermind from within your application. We will set up SSO credentials with a shared key that your servers can use to sign Single Sign-on access tokens.
2. Generate links to Mastermind within your application using your SSO credentials as described in the documentation below.

Generating SSO URLs

The SSO URLs to Mastermind follow the same format described in the "Linking to Mastermind" section of this document, with the addition of the following SSO parameters to be added to every link:

Name	Description
`provider`	The SSO partner ID assigned to your application by Genomenon.
`email`	The unique email address associated with the user's account in your application.
`first_name`	The user's first (given) name.
`last_name`	The user's last (family) name.
`organization`	The company or institution for which the user is affiliated.
`session_id`	The unique identifier used to indicate the user's current case or session within your application.
`timestamp`	The timestamp right now, in epoch (Unix) time. The time must be between five minutes behind and one minute ahead of the time on the Mastermind servers.
`token`	The Base-64 encoded SHA-256-digest-based HMAC hash of the URL request parameters

Example SSO URL

The initial SSO URL will be generated on your application's server using your Mastermind SSO secret key. This key should remain safe and secure on your application's servers and never be shared with anyone.

For this example, we'll assume your assigned `provider` ID, secret key, and the user's request parameters are as follows:

Parameter	Value
`disease`	`"melanoma"`
`gene`	`"braf"`
`variant`	`"v600e"`
`secret_key`	`"YourAppIsAwesome!"`
`provider`	`"yourapp"`
`email`	`"johnsmith@example.com"`
`first_name`	`"John"`
`last_name`	`"Smith"`
`organization`	`"The Lab"`
`session_id`	`"case-1234567"`
`timestamp`	`1551376523` (2019-02-28 12:55:23 EST converted to epoch time)
`token`	`"L2JhPqsSOYElxcyLlqofORR/cTePN7lIoSF7zy/+C+4="`

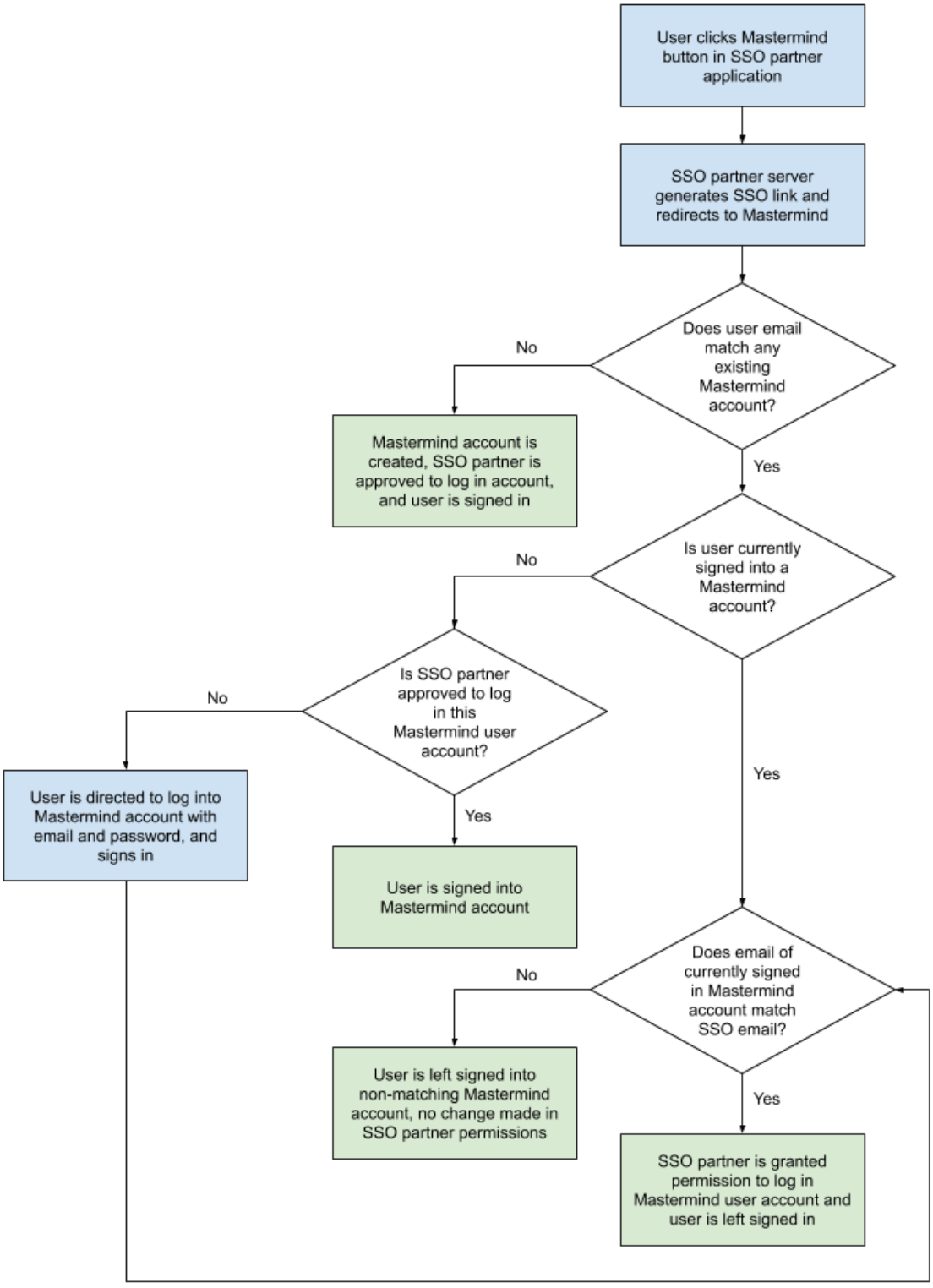
The token above was calculated by finding the HMAC of the concatenation of the request parameters using the secret_key. If your server uses the Ruby programming language, that would be done as follows:

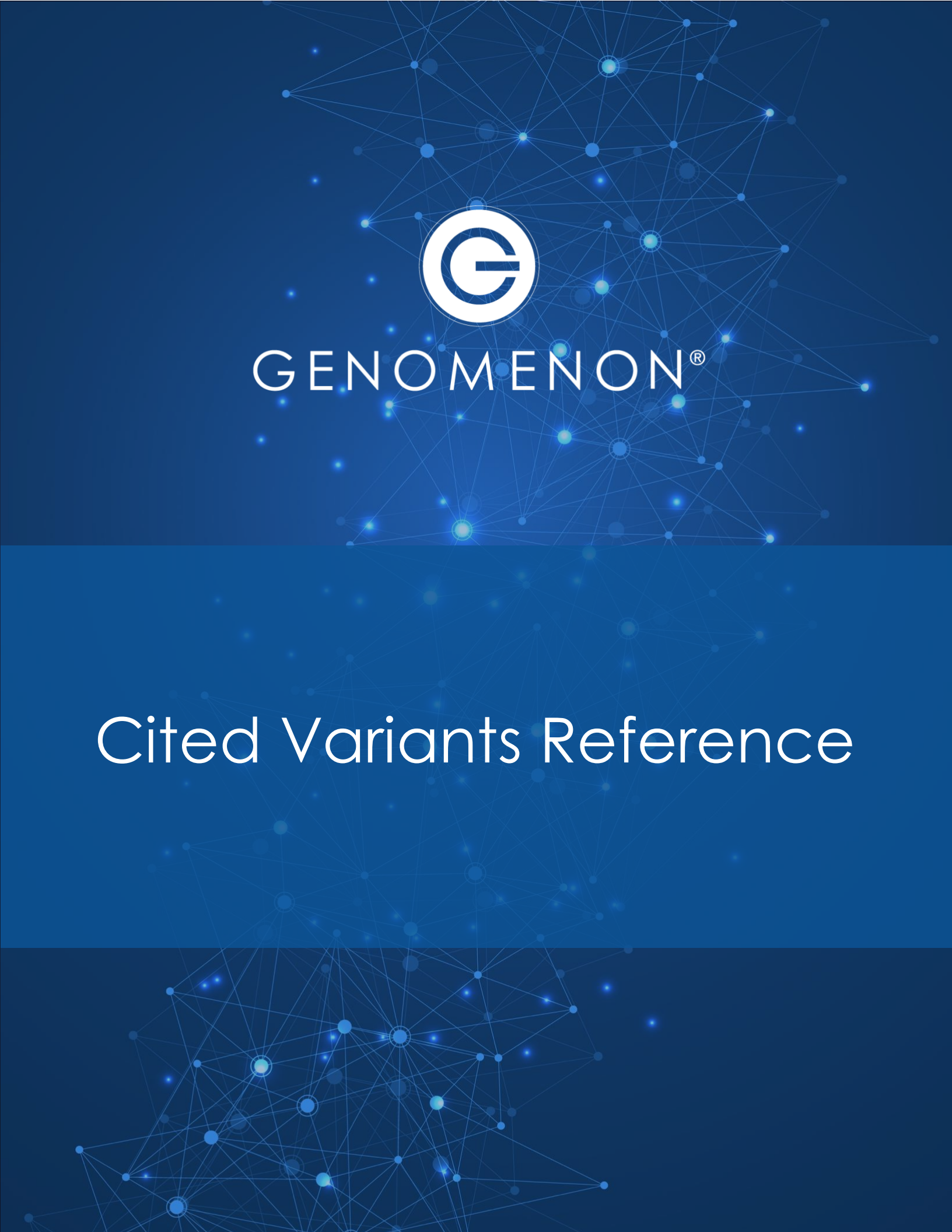
```
POST https://mastermind.genomenon.com/sso/accept\n
body:disease=melanoma&gene=braf&variant=v600eprovider=yourapp&email=johnsmith@example.com&first_name=Jo
hn&last_name=Smith&organization=The%20Lab&session_id=case-
1234567&timestamp=1551376523&token=L2JhPqsSOYElxcyLlqofORR%2FcTePN7lIoSF7zy%2F%2BC%2B4%3D
```

```
secret_key="YourAppIsAwesome!"
provider="yourapp"
email="johnsmith@example.com"
first_name="John"
last_name="Smith"
organization="The Lab"
session_id="case-1234567"
timestamp = Time.now.to_i #=> 1551376523

request_data = disease + gene + variant + provider + email + first_name + last_name + organization + session_id +
timestamp.to_s

token = Base64.encode64(OpenSSL::HMAC.digest(OpenSSL::Digest::Digest.new("sha256"), secret_key, request_data)).strip
```

The resulting SSO URL would then be a POST request from your server as follows:

The user experience workflow from the SSO partner application into Mastermind will follow this path, depending on the Mastermind user account matching the provided user email address in the request:





Cited Variants Reference

The Mastermind Cited Variants Reference file is a VCF file providing a snapshot of variant evidence available in Mastermind. Below is a truncated excerpt:

HGVS	MMCNT1	MMCNT2	MMCNT3	MMURL3
g.3209075C>T	4	4	4	https://mastermind.genomenon.com/…mutation=SLC4A11:Q812sa
g.865595A>G	1	1	1	https://mastermind.genomenon.com/…mutation=SAMD11:K45E
g.874491_874492delinsTA	0	0	1	https://mastermind.genomenon.com/…mutation=SAMD11:R168X

`MMCNT1` column (most specific)

cDNA-level exact matches. This is the number of articles that mention the variant at the nucleotide level in either the title/abstract or the full-text.

`MMCNT2` column

cDNA-level possible matches. This is the number of articles with nucleotide-level matches (from 1) plus articles with protein-level matches in which the publication did not specify the cDNA-level change, meaning they could be referring to this nucleotide-level variant but there is insufficient data in these articles to determine conclusively.

`MMCNT3` column (most sensitive)

This is the number of articles citing any variant resulting in the same biological effect as this variant. This includes the articles from MMCNT1 and MMCNT2 plus articles with alternative cDNA-level variants that result in the same protein effect.

`MMURL3` column

This is a deep-link into Mastermind for the selected variant, which shows all articles from MMCNT3, in order to investigate and explore the evidence in the literature.

If you are integrating the link URLs from the CVR into a customer-facing application, please contact us for instructions on properly embedding your own reference code into your Mastermind URLs to properly attribute users.

Current Version

To fetch the current version of the CVR in JSON format, automated periodic GET requests may be performed using the following URL. This will allow automatic detection of CVR updates to re-import the latest data available via the CVR:

```
curl https://mastermind.genomenon.com/cvr/version
```

The response will be JSON formatted. An example response is as follows:

```
{"version":"2019.03.14"}
```

Limitations

While the file does contain over 4.9 million variants seen in the medical literature, it doesn't include everything in Mastermind's ever-expanding database.

To summarize:

- Substitutions, intronic and splice-site variants, and UTR variants are in the reference file.
- Duplications, Deletions, Insertions, Indels, and Inversions may be in the reference file, depending on the complexity of the variation and the level of nomenclatures used within the literature. For these, we recommend querying the Mastermind API if they are not in the reference file, for maximum sensitivity.
- Frameshifts and upstream and downstream gene variants are not in the reference file, but can be queried in the API.

For more information about using the Cited Variants Reference and for licensing information, please see the Mastermind Cited Variants Reference download page.

GENOMENON®

Mastermind API Endpoints

The Mastermind API is a set of programmatic endpoints available to interact with the Mastermind data at a more detailed and granular level.

API Endpoints

`/suggestions` API endpoint

BASE API

Returns the canonical name(s) to query in Mastermind for the given disease, gene, or variant.

`/counts` API endpoint

BASE API

Returns the number of articles that match the given query.

`/articles` API endpoint

ADVANCED API

Returns the list of articles, ranked by relevance, for the given query.

`/diseases` API endpoint

ADVANCED API

Returns the top 5 diseases, ranked by number of matched articles, for the given query.

`/genes` API endpoint

ADVANCED API

Returns the top 5 genes, ranked by number of matched articles, for the given query.

`/variants` API endpoint

ADVANCED API

Returns the top 5 variants, ranked by number of matched articles, for the given query.

`/article_info` API endpoint

ADVANCED API

Returns the matched diseases, genes, variants, and article meta-data for the given article PMID.

`/file_annotations/counts` API endpoint

ADVANCED API

Returns an annotated VCF from an input VCF file containing multiple variants as genomic coordinates with each variant in the output file annotated with the number of articles and Mastermind URL for each variant.

Authentication

All API requests to both Basic and Advanced endpoints must be authenticated using your API token. Simply pass the API token as the `X-API-TOKEN` header in each request. An example request to the `/counts` endpoint would look like this, using Curl from the command line:

```
curl -H "Content-type: application/json" -H "X-API-TOKEN: [[your API token here]]" -X GET
"https://mastermind.genomenon.com/api/v2/counts?&disease=melanoma&gene=braf&variant=braf:v600e"
```

If you do not know your API token, you can fetch it by visiting the API homepage and clicking the link labeled "Click here to fetch your API token."

If you do not have an API token assigned, you may reach out to sales@genomenon.com.

Request Prefix

All of the following endpoints should be prefixed with the current version of the API, which is v2:

```
https://mastermind.genomenon.com/api/v2
```

So, for example, the `/suggestions` endpoint would be queried via:

```
https://mastermind.genomenon.com/api/v2/suggestions
```

Request Type

As all endpoints in the Mastermind API V2 are to retrieve data, all API requests should be `GET` requests (not `POST` or other type).

URL-encoding Query Values

All parameter values to the API should be encoded. For example, querying the `/suggestions` endpoint for "non-small cell lung cancer" would be encoded as follows:

```
https://mastermind.genomenon.com/api/v2/suggestions?disease=non-small%20cell%20lung%20cancer
```

Likewise, querying the `/suggestions` endpoint for "braf:c.1799t>a" would be encoded as:

```
https://mastermind.genomenon.com/api/v2/suggestions?variant=braf%3Ac.1799t%3Ea
```

Multiple parameter values

Some endpoint parameter input values, such as `categories`, allow multiple values to be input simultaneously, These are shown in the parameter documentation as array values, such as:

```
["in vivo", "de novo"]
```

For `GET` requests, these are specified by repeating the parameter name appended with square brackets in the URL with the different values, such as:

```
https://mastermind.genomenon.com/api/v2/counts?gene=braf&categories[]=in%20vivo&categories[]=de%20novo
```

Code Examples

For live code examples using the Mastermind API, see the following repository on Github:

https://github.com/Genomenon/mastermind-api-clients

The `/suggestions` API endpoint returns the canonical name(s) to query in Mastermind for the given disease, gene, or variant.

Query Parameters

Name	Description	Example
`disease`	The disease name you would like to look up.	`"non-small cell lung cancer"`
`gene`	The gene name to be looked up.	`"v-Raf murine sarcoma viral oncogene homolog B"`
`variant`	The variant to be looked up, which may be specified in HGVS cDNA (c.) nomenclature, protein (p.) nomenclature, or rsID. When querying by cDNA or protein nomenclatures, be sure to include the gene symbol.	`"braf:c.1799t>a"`

Response Format

```
{
  "type":      [["disease", "gene", or "variant"]],
  "matched":   [[synonym or nomenclature matched]],
  "canonical": [[canonical name to use for Mastermind queries]],
  "url":       [[link to Mastermind query using the canonical name]]
}
```

Example Response

```
{
  "type": "disease",
  "matched": "acute myeloid leukemia",
  "canonical": "leukemia, myeloid, acute",
  "url": "https://mastermind.genomenon.com/search?disease=leukemia,%20myeloid,%20acute"
}
```

The `/counts` API endpoint returns the number of articles that match the given query.

Query Parameters

Name	Description	Example
`disease`	The disease name associated with the articles of interest, specified in lowercased MeSH terminology.	`"melanoma"`
`gene`	The gene symbol associated with the articles of interest, specified in lowercase by their HGNC gene symbol.	`"braf"`
`variant`	The variant associated with the articles of interest, specified in lowercase by their protein effect using HGVS protein-level nomenclature for SNVs, or modified nomenclature for other variant types. Use the `/suggestions` endpoint to get the queryable variant nomenclature for your variant.	`"braf:v600e"`
`categories`	The Mastermind categories and keywords associated with the articles of interest, in lowercase. These should be chosen from the category/keyword list found in Mastermind, including ACMG/AMP criteria and other clinical significance indicators. *NOTE: Passing multiple categories will **increase** the scope of the search, as superset of articles will be returned that match **any** of the specified categories.*	`["in vivo", "de novo"]` accepts multiple values in same query)
`scope`	The scope of the article search, from most specific PubMed-only searches, to more sensitive search including the full-text, to most sensitive including supplemental materials.	`"abstract", "fulltext", "supplemental" (default)`
`since`	The starting date from which to return matches, as an Epoch Unix timestamp. When specified, only results matched since the provided date are returned. This can be used for example as part of a weekly cronjob to query for all new results since the last time the endpoint was queried for data.	`1556918275` (for May 3, 2019 9:17:55 PM GMT)

Response Format

```
{
  "article_count": [[total number of articles]]
  "url": [[link to Mastermind results]]
}
```

Example Response

```
{
  "article_count": 4613,
  "url": "https://mastermind.genomenon.com/detail?disease=melanoma&gene=braf&mutation=braf:v600e"
}
```

The `/articles` API endpoint returns the first 5 PMIDs, ranked by relevance, for the given query.

Query Parameters

See "Counts" endpoint. Addiitonal parameters for this endpoint are:

Name	Description	Example
`page`	The desired page of results to return in the "articles" results array.	2

Response Format

```
{
  "article_count": [[total number of articles]],
  "pages": [[number of results pages available]],
  "page": [[current results page]],
  "articles": [
    {
      "pmid": [[pmid]],
      "title": [[title]],
      "publication_date": [[normalized publication date]],
      "journal": [[journal ISO code]],
      "authors": [[list of author names]],
      "url": [[link to view article in Mastermind]]
    }
  ],
```

Example Response (truncated to first article for clarity)

```
{
  "article_count": 8392,
  "pages": 1679,
  "page": 1,
  "articles": [
    {
      "pmid": "23543365",
      "title": "Gossypin as a novel selective dual inhibitor of V-RAF murine sarcoma viral oncogene
homolog B1 and cyclin-dependent kinase 4 for melanoma.",
      "journal": "Mol. Cancer Ther.",
      "publication_date": "2013-03-29",
      "authors": [
        "Bhaskaran S",
        "Dileep KV",
        "Deepa SS",
        "Sadasivan C",
        "Klausner M",
        "Krishnegowda NK",
        "Tekmal RR",
        "VandeBerg JL",
        "Nair HB"
      ],
      "url": "https://mastermind-staging.genomenon.com/detail?disease=melanoma&gene=braf&pmid=23543365"
    }
  ],
  "url": "https://mastermind-staging.genomenon.com/detail?disease=melanoma&gene=braf"
```

The `/diseases` API endpoint returns the top 5 diseases, ranked by number of matched articles for the given query.

Query Parameters

See "Counts" endpoint

Response Format

```
{
  "article_count": [[total number of articles]],
  "disease_count": [[total number of diseases]],
  "diseases": [
    {
      "key": [[MeSH disease name]],
      "article_count": [[number of articles for disease and given query]],
      "url": [[link to Mastermind for disease and given query]]
    }
  ],
  "url": [[link to Mastermind query for the given gene and variant]]
}
```

Example Response

```
{
  "article_count": 35647,
  "disease_count": 2360,
  "diseases": [
    {
      "key": "neoplasms",
      "article_count": 21640,
      "url": "https://mastermind.genomenon.com/detail?disease=neoplasms&gene=braf"
    },
    {
      "key": "melanoma",
      "article_count": 8392,
      "url": "https://mastermind.genomenon.com/detail?disease=melanoma&gene=braf"
    },
    {
      "key": "carcinoma",
      "article_count": 8144,
      "url": "https://mastermind.genomenon.com/detail?disease=carcinoma&gene=braf"
    },
    {
      "key": "neoplasm metastasis",
      "article_count": 5833,
      "url": "https://mastermind.genomenon.com/detail?disease=neoplasm+metastasis&gene=braf"
    },
    {
      "key": "thyroiditis",
      "article_count": 3957,
      "url": "https://mastermind.genomenon.com/detail?disease=thyroiditis&gene=braf"
    }
  ],
  "url": "https://mastermind.genomenon.com/search?gene=braf"
}
```

The `/genes` API endpoint returns the top 5 genes, ranked by number of matched articles, for the given query.

Query Parameters

See "Counts" endpoint

Response Format

```
{
  "article_count": [[total number of articles]],
  "gene_count": [[total number of genes]],
  "genes": [
    {
      "symbol": [[gene symbol]],
      "article_count": [[number of articles for gene and given query]],
      "url": [[link to Mastermind for gene and given query]]
    }
  ],
  "url": [[link to Mastermind query for the given disease]]
}
```

Example Response

```
{
  "article_count": 67028,
  "gene_count": 18362,
  "genes": [
    {
      "symbol": "IL2",
      "article_count": 9570,
      "url": "https://mastermind.genomenon.com/detail?disease=melanoma&gene=IL2"
    },
    {
      "symbol": "CD8A",
      "article_count": 8770,
      "url": "https://mastermind.genomenon.com/detail?disease=melanoma&gene=CD8A"
    },
    {
      "symbol": "BRAF",
      "article_count": 8392,
      "url": "https://mastermind.genomenon.com/detail?disease=melanoma&gene=BRAF"
    },
    {
      "symbol": "CES2",
      "article_count": 8383,
      "url": "https://mastermind.genomenon.com/detail?disease=melanoma&gene=CES2"
    },
    {
      "symbol": "FBRS",
      "article_count": 8225,
      "url": "https://mastermind.genomenon.com/detail?disease=melanoma&gene=FBRS"
    }
  ],
  "url": "https://mastermind.genomenon.com/search?disease=melanoma"
}
```

The `/variants` API endpoint returns the top 5 variants, ranked by number of matched articles, for the given query.

Query Parameters

See "Counts" endpoint

Response Format

```
{
  "article_count": [[total number of articles]],
  "variant_count": [[total number of variants]],
  "variants": [
    {
      "key": [[Mastermind variant key name]]},
      "gene": [[HGVS gene symbol]],
      "article_count": [[number of articles for variant and given query]],
      "url": [[link to Mastermind for variant and given query]]
    }
  ],
  "url": [[link to Mastermind query for the given disease]]
}
```

Example Response

```
{
  "article_count": 32203,
  "variant_count": 1305,
  "variants": [
    {
      "gene": "pah",
      "key": "pah:R408W",
      "article_count": 409,
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=pah&mutation=pah:R408W"
    },
    {
      "gene": "pah",
      "key": "pah:R261Q",
      "article_count": 353,
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=pah&mutation=pah:R261Q"
    },
    {
      "gene": "pah",
      "key": "pah:Y414C",
      "article_count": 245,
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=pah&mutation=pah:Y414C"
    },
    {
      "gene": "pah",
      "key": "pah:R158Q",
      "article_count": 234,
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=pah&mutation=pah:R158Q"
    },
    {
      "gene": "pah",
      "key": "pah:P281L",
      "article_count": 230,
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=pah&mutation=pah:P281L"
    }
  ],
  "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=pah"
}
```

The `/article_info` API endpoint returns the matched diseases, genes, variants, and article meta-data for a given article PMID.

Query Parameters

Name	Description	Example
`pmid`	The PubMed ID for the desired article.	`23543365`

Response Format

```
{
  "pmid": [[PMID of article]],
  "title": [[PubMed title]],
  "abstract": [[list of abstract parts, each with "index", "text", and "label"]]
  "journal": [[journal ISO abbreviation]],
  "publication_date": [[normalized publication date]],
  "doi": [[DOI from PubMed]],
  "authors": [[list of author names]],
  "mesh_headings": [[list of MeSH heading names from PubMed]],
  "substances": [[list of substance names from PubMed]],
  "gene_symbols": [[list of gene symbols from PubMed]],
  "included_in_mastermind": [[true/false, whether the article is shown in Mastermind]],
  "pubmed_indexed": [[true/false, whether Mastermind has indexed the PubMed data for article]],
  "fulltext_indexed": [[true/false, whether Mastermind has indexed the full-text for article]],
  "supplemental_data_indexed": [[true/false, whether Mastermind has indexed the supplemental data for
article]],
  "english_language": [[true/false, whether article is English-language]],
  "diseases": [
    {
      "key": [[MeSH disease name]],
      "url": [[link to Mastermind query for disease]]
    }
  ],
  "genes": [
    {
      "symbol": [[HGVS gene symbol]],
      "mentions": [[number of gene mentions in article]],
      "url": [[link to Mastermind for gene]],
      "variants": [
        {
          "key": [[Mastermind variant key name]],
          "url": [[link to Mastermind for variant]]
        }
      ]
    }
  ],
  "url": [[link to Mastermind query for the article]]
}
```

Example Response

See next page

Example Response (truncated results)


```
{
  "pmid": "23543365",
  "title": "Gossypin as a novel selective dual inhibitor of V-RAF murine sarcoma viral oncogene homolog
B1 and cyclin-dependent kinase 4 for melanoma.",
  "abstract": [
    {
      "index": 0,
      "text": "Mutation in the BRAF gene (BRAFV600E) exists in nearly 70% of human melanomas…"
      "label": ""
    }
  ],
  "journal": "Mol. Cancer Ther.",
  "publication_date": "2013-03-29",
  "doi": "10.1158/1535-7163.MCT-12-0965",
  "authors": ["Bhaskaran S", "Dileep KV", "Deepa SS", "Sadasivan C", "Klausner M", "Krishnegowda NK",
"Tekmal RR", "VandeBerg JL", "Nair HB"],
  "mesh_headings": ["Animals", "Cell Cycle", "Cell Line, Tumor", "Cell Movement", "Cyclin-Dependent
Kinase 4", "Disease Models, Animal", "Female", "Flavonoids", "Humans", "MAP Kinase Signaling System",
"Melanoma", "Mice", "Molecular Docking Simulation", "Mutation", "Protein Binding", "Proto-Oncogene
Proteins B-raf", "Transplantation, Heterologous", "Tumor Burden"],
  "substances": [ "Flavonoids", "gossypin", "Proto-Oncogene Proteins B-raf", "Cyclin-Dependent Kinase
4"],
  "included_in_mastermind": true,
  "pubmed_indexed": true,
  "fulltext_indexed": true,
  "supplemental_data_indexed": false,
  "english_language": true,
  "diseases": [
    {
      "key": "retinoblastoma",
      "url": "https://mastermind.genomenon.com/search?disease=retinoblastoma"
    },
    …
  ],
  "genes": [
    {
      "symbol": "BRAF",
      "mentions": 44,
      "variants": [
        {
          "key": "V600E",
          "url":
"https://mastermind.genomenon.com/detail?disease=all+diseases&gene=braf&mutation=braf:V600E"
        }
      ],
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=braf"
    },{
      "symbol": "EPHB2",
      "mentions": 31,
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=ephb2"
    }, {
      "symbol": "RB1",
      "mentions": 2,
      "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=rb1"
    },
    …
  ],
  "url": "https://mastermind.genomenon.com/detail?disease=all+diseases&gene=cdk4&pmid=23543365"
}
```

The `/file_annotations/counts` API endpoint provides bulk counts for variants by returning an annotated VCF from an input VCF file containing multiple variants as genomic coordinates with each variant in the output file annotated with the number of articles and Mastermind URL for each variant.

This process is made up of four sequential endpoints to ease integration from different workflows and allow robust handling of potentially large files. The four steps are:

1. Create a new bulk annotation process
2. Upload the VCF data
3. Check the status of the annotation process
4. Download the annotated VCF file when ready

1. Create a new bulk annotation process

This is a `POST` request to `/file_annotations/counts`. The response will create a unique identifier for your bulk annotation job and respond with a URL for submitting the VCF data as well as the URL endpoint to check the status of the annotation process.

Query Parameters

Name	Description	Example
`assembly`	The build version used to produce the VCF file, either `grch37` or `grch38`.	`grch37`
`filename`	The name of the input VCF file to upload.	`my_file.vcf.gz`

Example Request

```
curl -XPOST -L \
-H "X-API-TOKEN: [[your API token here]]" \
"https://mastermind.genomenon.com/api/v2/file_annotations/counts?filename=my_file.vcf.gz&assembly=grch37"
```

Response Format

```
{
  "job_id": [[unique annotation job identifier]],
  "state": [[new annotation job status]],
  "assembly": [[assembly used for file]],
  "upload_url": [[URL to upload VCF file]],
  "input_filename": [[file name of input file]],
  "created_at": [[timestamp of when job was created]],
  "job_url": [[URL to check status of annotation job]]
}
```

Example Response

```
{
  "job_id": "ff91654c-5ad4-4f1c-9ed3-60a8492ce689",
  "state": "created",
  "assembly": "GRCh37",
  "upload_url": "https://genomenon-vcf-in.s3.us-west-2.amazonaws.com/…",
  "input_filename": "my_file.vcf.gz",
  "created_at": "2019-08-29T08:37:11.081Z",
  "job_url": "https://mastermind.genomenon.com/api/v2/file_annotations/counts/ff91654c-5ad4…"
}
```

2. Upload the VCF data

This is a `PUT` request to the URL provided in the `upload_url` attribute of the response of step 1. This will respond with a 200 status code once the upload is complete.

Example Request

```
curl -X PUT --data-binary @/path/to/my_file.vcf -L \
   -H "X-API-TOKEN: [[your API token here]]" \
   -H "Content-Type: application/octet-stream" \
"https://genomenon-vcf-in.s3.us-west-2.amazonaws.com/…"
```

3. Check the status of the annotation process

This is a `GET` request to `/file_annotations/counts/[[job ID]]` endpoint provided in the `job_url` attribute in the response of step 1. The response will include the status of the job, and once complete will provide the download URL.

Response Format

```
{
  "job_id": [[unique annotation job identifier]],
  "state": [[new annotation job status]],
  "assembly": [[assembly used for file]],
  "input_filename": [[file name of input file]],
  "download_url": [[URL to upload VCF file]],
  "input_size": [[size of input file in bytes]],
  "input_etag": [[eTag for input file]],
  "output_filename": [[file name of output file]],
  "output_size": [[size of output file in bytes]],
  "output_etag": [[eTag for output file]],
  "created_at": [[timestamp of when job was created]],
  "started_at": [[timestamp of when annotation process was started]],
  "completed_at": [[timestamp of when annotation process was finished]],
  "elapsed": [[elapsed time in seconds]],
  "records": [[number of records in file]],
  "annotated": [[number of records annotated in file]],
  "job_url": [[URL to check status of annotation job]]
}
```

Example Response

```
{
  "job_id": "ff91654c-5ad4-4f1c-9ed3-60a8492ce689",
  "state": "succeeded",
  "assembly": "GRCh37",
  "input_filename": "my_file.vcf.gz",
  "download_url": "https://mastermind.genomenon.com/api/v2/file_annotations/counts/ff91654c-5ad4…",
  "input_size": 325103,
  "input_etag": "\"087450a21b0bffc946b716f9f18f73cf\"",
  "output_filename": "my_file.vcf.gz",
  "output_size": 431216,
  "output_etag": "\"321b3e7233cdbed6d0821083a1b14e47\"",
  "created_at": "2019-08-29T08:37:11.081Z",
  "started_at": "2019-08-29T08:37:22.649Z",
  "completed_at": "2019-08-29T08:47:32.716Z",
  "elapsed": 284,
  "records": 6980,
  "annotated": 4623,
  "job_url": "https://mastermind.genomenon.com/api/v2/file_annotations/counts/ff91654c-5ad4…"
}
```

4. Download the annotated VCF file when ready

This is a `GET` request to `/file_annotations/counts/[[job ID]]/download` endpoint provided in the `download_url` attribute in the response of step 3. The response will be a gzip-compressed annotated VCF file.

The statuses possible from the `state` attribute returned by the endpoint in step 3 are:

- `created`
- `started`
- `succeeded`
- `failed`

The output and completion attributes will show up once the state has been updated to succeeded. This is also when the download_url will be returned, at which point the file may be downloaded.

Example Annotation

```
#CHROM  POS     ID              REF     ALT     QUAL    FILTER  INFO
1       2160390 rs28384811      C       G       .       PASS
MMCNT=3;MMURI=https://mastermind.genomenon.com/detail?disease=all%20diseases&gene=SKI&mutation=SKI:A
```

200: Success

When querying endpoints, successful responses will return an HTTP response code of `200` (success).

400: Bad Request

Queries lacking an API token will return an HTTP response code of `400` (bad response). A JSON body containing an error key will also be returned explaining what cased the issue. Some issues, such as querying an endpoint without URL-encoding parameter values may cause a `400` response without a JSON body.

401: Unauthorized

Queries to API endpoints not granted to your API subscription, using an inactive or expired API token, or hitting your overall annual query limits, will return an HTTP response code of `401` (unauthorized). A JSON body containing an `error` key will also be returned explaining what caused the issue.

404: Not Found

If a given disease, gene, or variant input are not found, the API will return an HTTP response code of `404` (not found). A JSON body containing an `errors` key will also be returned, including which input(s) were not recognized, for example:

```
{
  "errors": [
    {
      "message": "specified variant 'blah' not found",
      "parameter": "variant"
    }
  ]
}
```

If this happens, considering querying the `/suggestions` endpoint with your disease, gene, or variant to get the appropriate value for querying the other Mastermind API endpoints.

422: Unprocessable Entity

Invalid queries, such as querying an endpoint with required fields omitted, will return an HTTP response code of `422` (unprocessable entity). A JSON body containing an `errors` key will also be returned explaining what caused the issue, for example:

```
{
  "errors": [
    {
      "message": "at least one parameter (disease, gene, or variant) must be specified"
    }
  ]
}
```

429: Too Many Requests

Queries which exceed the per-second or per-day throughput rate limits will temporarily return an HTTP response code of `429` (too many requests). If this happens, wait or throttle requests before trying again.