
VDJdb server Documentation

Release 1.0.1

Dmitry Bagaev, Mikhail Shugay

Jan 16, 2018

Contents

1	Table of Contents	3
1.1	Installing VDJdb server	3
1.2	VDJdb server REST API [BETA]	4

The **VDJdb-server** application implements a browser-based GUI for the VDJdb, a database of T-cell receptor sequences with known antigen specificities.

This documentation describes the installation and setup of VDJdb-server at your local network, as well as the REST API for the **VDJdb** portal.

1.1 Installing VDJdb server

First make sure that you have installed Java Runtime Environment (JRE) v1.8 by running `java -version`. Any recent Linux distribution will provide it via its package manager. If not, or if your system is running MacOSX or Windows, download the JRE from [Oracle](#).

You should also download and compile the VDJtools and VDJdb-standalone software using `mvn clean install` and `gradle clean build` respectively. Please check that the versions of VDJtools and VDJdb-standalone are matched to that in `.sbt` file coming with VDJdb-server.

1.1.1 Installing binaries

This section is intended for advanced users. We recommend using the standalone VDJdb application with command line interface that can be found [here](#).

The most straightforward way to install VDJdb as a local server is to download the [latest release package](#).

After downloading unzip the package wherever you want, but please avoid long paths and spaces (Windows version is especially sensitive to it).

You can find the server executable in `bin/` directory. To set up the server:

- Run `vdjdb.bat` file (Windows)
- Run `bash vdjdb -Dconfig.file=../conf/application.conf` in your console (Linux/Mac OS)

Wait until the server is started, and go to `localhost:9000` URL in your browser to open VDJviz.

To stop application just press `Ctrl-C` at any time in console.

Note: Note that an exception will be thrown in case the 9000 port is busy: `org.jboss.netty.channel.ChannelException: Failed to bind to: /0.0.0.0:9000`. In order to fix it, either close the application that is using this port (in UNIX the `lsof -i:9000` will give the processes that are using the port) or pass

the `-Dhttp.port=XXXX` (where XXXX is new port id) argument to `vdjviz` shell script (UNIX) / `vdjviz.bat` (Windows)

1.1.2 Configuration

VDJdb server configuration can be performed by manually editing `application.conf` file in the `conf/` directory. The configuration file has the following fields:

`application.secret` The secret key used in cryptographic hash functions.

`uploadPath` Specifies the path that will be used by VDJviz to store user's uploaded files. You can use `'~'` symbol as a shortcut for user home directory. Default: `/tmp`

`maxFileSize` File size limit in kB Default: 0 (no limit)

`maxFilesCount` Limit on the number of uploaded files per user. Default: 0 (no limit)

`deleteAfter` Time period after which uploaded files are deleted from the server, in hours. Default: 0 (never)

`automaticDatabaseUpdate` Default: `true`

`useLocalDatabase` Default: `false`

`databasePath` Default: `~/database/`

`db.default.url` Points to the path that will be used to store H2 database file. Default: `~/vdjdb/h2.db`

Important: Standalone version uses [H2 Database](#) for handling metadata by default, if you want to change H2 to another DBMS please see the corresponding [Play documentation section](#) You can also use this database to manually modify user limits.

`securesocial.*` Refer to [Secure social](#) configuration.

`smtp.*` Refer to [Play! Framework SMTP](#) server configuration.

1.2 VDJdb server REST API [BETA]

1.2.1 Accessing metadata

Information on database columns (such as their IDs, names and other metadata) can be obtaining by performing a GET request to `https://vdjdb.cdr3.net/search/columns`.

For example,

```
curl -X GET https://vdjdb.cdr3.net/search/columns
```

will return a JSON object with the following structure.

```
[
  ...
  {
    "name": "gene",
    "metadata" : {
      "columnType": "txt",
      "visible" : "1",
```



```

        "searchable" : "1"
        "dataType" : "factor",
        "title": "Gene",
        "comment" : "TCR chain: alpha or beta.",
    },
    "autocomplete" : true,
    "values" : [ "TRA" , "TRB" ]
}
...
]
```

1.2.2 Searching the database

You can query the database by sending a POST request with a specific JSON content to <https://vdjdb.cdr3.net/search>.

The structure of JSON query is the following:

```
{
  "textFilters": [
    ...
  ],
  "sequenceFilters": [
    ...
  ]
}
```

Where the **textFilters** structure is

```
{
  "columnId": "(string)",      // Column name (any available column)
  "value": "(string)",        // Search value
  "filterType": "(string)",    // Filter type: exact, exact_set, pattern, substring_
  ↪set, level
  "negative": (boolean)        // Return only results that do not match the filter
}
```

and the **sequenceFilters** structure is

```
{
  "columnId": "(string)",      // Column name (cdr3 or antigen.epitope)
  "query": "(string)",        // Search query
  "substitutions": (int),      // The number of substitutions
  "insertions": (int),         // The number of insertions
  "deletions": (int),          // The number of deletions
  "total": (int)               // Total number of mutations allowed
}
```

Sequence filters can only be applied to columns with "columnType": "seq" and will invoke a search and alignment procedure (more precisely, a sequence tree search). Text filters can be specified for any available column.

Note: Filters mirror the filtering functions implemented in VDJdb-standalone. Thus, parameter description can be found in the documentation of `*Filter.groovy` classes implemented for **sequence** and **text** columns respectively.

The structure of JSON response is the following:

```
[
  ...
  {
    "entries": [
      ...
      { "columnName": "gene", "value": "TRA" }
      ...
    ]
  }
  ...
]
```

For example, the following request

```
curl -H "Content-Type: application/json" -X POST -d '{ "textFilters" : [{"columnId":
↪ "cdr3", "value": "CAAAASGGSYIPTF", "filterType": "exact", "negative": false }],
↪ "sequenceFilters" : [] }' https://vdjdb.cdr3.net/search
```

will produce

```
[{
  "entries": [{
    "columnName": "complex.id",
    "value": "131"
  }, {
    "columnName": "gene",
    "value": "TRA"
  }, {
    "columnName": "cdr3",
    "value": "CAAAASGGSYIPTF"
  }, {
    "columnName": "v.segm",
    "value": "TRAV1-2*01"
  }, {
    "columnName": "j.segm",
    "value": "TRAJ6*01"
  }, {
    "columnName": "species",
    "value": "HomoSapiens"
  }, {
    "columnName": "mhc.a",
    "value": "HLA-B*35:01"
  }, {
    "columnName": "mhc.b",
    "value": "B2M"
  }, {
    "columnName": "mhc.class",
    "value": "MHCI"
  }, {
    "columnName": "antigen.epitope",
    "value": "EPLPQGQLTAY"
  }, {
    "columnName": "antigen.gene",
    "value": "BZLF1"
  }, {
    "columnName": "antigen.species",
    "value": "EBV"
  }, {
```

```

        "columnName": "reference.id",
        "value": "PMID:16148129"
    }, {
        "columnName": "method",
        "value": "{\\"frequency\\": \\"4/4\\", \\"identification\\": \\"antigen-loaded-
↪targets\\", \\"sequencing\\": \\"sanger\\", \\"singlecell\\": \\"\\", \\"verification\\": \
↪"antigen-loaded-targets,tetramer-stain\\"}"
    }, {
        "columnName": "meta",
        "value": "{\\"cell.subset\\": \\"CD8+\\", \\"clone.id\\": \\"MW2\\", \\"donor.MHC\\": \
↪"HLA-B*35:01\\", \\"donor.MHC.method\\": \\"sequencing\\", \\"epitope.id\\": \\"\\", \
↪"replica.id\\": \\"\\", \\"samples.found\\": 1, \\"structure.id\\": \\"\\", \\"studies.found\\
↪": 1, \\"study.id\\": \\"\\", \\"subject.cohort\\": \\"healthy\\", \\"subject.id\\": \\"\\", \
↪"tissue\\": \\"CTL culture\\"}"
    }, {
        "columnName": "cdr3fix",
        "value": "{\\"cdr3\\": \\"CAAAASGGSYIPTF\\", \\"cdr3_old\\": \\"CAAAASGGSYIPTF\\", \
↪"fixNeeded\\": false, \\"good\\": true, \\"jCanonical\\": true, \\"jFixType\\": \
↪"NoFixNeeded\\", \\"jId\\": \\"TRAJ6*01\\", \\"jStart\\": 4, \\"vCanonical\\": true, \\"vEnd\\
↪": 2, \\"vFixType\\": \\"NoFixNeeded\\", \\"vId\\": \\"TRAV1-2*01\\"}"
    }, {
        "columnName": "vdjdb.score",
        "value": "3"
    }, {
        "columnName": "web.method",
        "value": "culture"
    }, {
        "columnName": "web.method.seq",
        "value": "sanger"
    }, {
        "columnName": "web.cdr3fix.nc",
        "value": "no"
    }, {
        "columnName": "web.cdr3fix.unmp",
        "value": "no"
    }
}
}]

```

Note: Column description can be found [here](#).

Warning: The columns `method`, `meta`, `cdr3fix`, `web.method`, `web.method.seq` are likely to be removed in the future.