HOWARD Help Configuration

Contents

1 Introduction

HOWARD Configuration JSON file defined default configuration regarding resources (e.g. threads, memory), settings (e.g. verbosity, temporary files), default folders (e.g. for databases) and paths to external tools.

Examples:

Example of a configuration JSON file

```
{
  "threads": 8,
  "memory": null,
  "verbosity": "WARNING",
  "folders": {
    "databases": {
      "genomes": "~/howard/databases/genomes/current",
      "annotations": [
        "~/howard/databases/annotations/current",
        "~/howard/databases/dbnsfp/current",
        "~/howard/databases/dbsnp/current"
      ],
      "parquet": [
        "~/howard/databases/annotations/current"
      ],
      "bcftools": [
        "~/howard/databases/annotations/current"
      ],
      "annovar": "~/howard/databases/annovar/current",
      "snpeff": "~/howard/databases/snpeff/current",
      "varank": "~/howard/databases/varank/current"
    }
  },
  "tools": {
    "bcftools": "bcftools",
    "bgzip": "bgzip",
    "java": "java",
    "snpeff": "~/howard/tools/snpeff/current/bin/snpEff.jar",
    "annovar": "~/howard/tools/annovar/current/bin/table_annovar.pl",
    "exomiser": "~/howard/tools/exomiser/current/bin/exomiser-cli-13.2.0.jar",
    "splice": {
       "docker": {
         "image": "bioinfochrustrasbourg/splice:0.2.1",
         "entrypoint": "/bin/bash",
         "options": null,
         "command": null
       }
    }
  }
}
```


2 folders

Folders configuration such as for databases.

2.1 databases

Default folders for databases that follows the specific database HOWARD format. These folders will be used in HOWARD tools to autodetect databases by their name and using assembly. Within database folders, multiple releases can be provides (e.g. 'b152' and 'b156' for dbSNP and 'hg19' assembly within folder '~/howard/databases/dbsnp/current/hg19', in 2 subfolders resp.)

Format: `/path/to/databases/<db_name>/<db_release>/<assembly>/<database_files>`

Examples:

Example of a configuration for databases folders


```
{
   "databases": {
```

```
        "genomes": "~/howard/databases/genomes/current",
        "annotations": [
            "~/howard/databases/annotations/current",
            "~/howard/databases/dbnsfp/current",
            "~/howard/databases/dbsnp/current"
        ],
        "parquet": [
            "~/howard/databases/annotations/current"
        ],
        "bcftools": [
            "~/howard/databases/bcftools/current"
        ],
        "annovar": "~/howard/databases/annovar/current",
        "snpeff": "~/howard/databases/snpeff/current",
        "exomiser": "~/howard/databases/exomiser/current"
    }
}
```

2.1.1 genomes

Genome folder with, for each assembly, FASTA files, indexes, and all files generated by pygenome module.

Type: `Path`

Format: `A folder path (without assembly)`

Default: `~/howard/databases/genomes/current`

Examples:

Path to genomes folder

```
{
    "genomes": "~/howard/databases/genomes/current"
}
```

2.1.2 annotations

Annotation databases folders that contains databases in various format such as Parquet, VCF, duckDB and TSV.

Type: `Path`

Format: `a list of folder path (without assembly)`

Default: `["~/howard/databases/annotations/current"]`

Examples:

Uniq folder with multiple annotations for Parquet annotation method, or other External tools

```
{
    "annotations": [
        "~/howard/databases/annotations/current"
    ]
}
```

Combinason of 2 folders with multiple annotations for Parquet annotation method, or other External tools

```
{
    "annotations": [
        "~/howard/databases/annotations/current",
        "~/howard/databases/dejavu/current",
        "~/howard/databases/dbnsfp/current"
    ]
}
```

2.1.3 parquet

Annotation databases folders that contains databases in Parquet format.

Format: `a list of folder path (without assembly)`

Default: `["~/howard/databases/annotations/current"]`

Examples:

Uniq folder with multiple annotations for Parquet annotation method

```
{
   "annotations": [
      "~/howard/databases/annotations/current"
   ]
}
```

Combinason of 2 folders with multiple annotations for Parquet annotation method

```
{
   "annotations": [
      "~/howard/databases/annotations/current",
      "~/howard/databases/dejavu/current",
      "~/howard/databases/dbnsfp/current"
   ]
}
```

2.1.4 bcftools

Annotation databases folders for BCFTools annotation.

Format: `a list of folder path (without assembly)`

Default: `["~/howard/databases/bcftools/current"]`

Examples:

Uniq folder with multiple VCF and BED files for BCFTools annotation

```
{
   "bcftools": [
      "~/howard/databases/bcftools/current"
   ]
}
```

Combinason of 2 folders with multiple VCF and BED files for BCFTools annotation

```
{
   "bcftools": [
      "~/howard/databases/bcftools/current",
      "~/howard/databases/dejavu/current"
   ]
}
```

2.1.5 annovar

Annotation databases folder for Annovar annotation.

Format: `a list of folder path (without assembly)`

Default: `["~/howard/databases/annovar/current"]`

Examples:

Uniq folder with multiple Annovar TXT files for Annovar annotation

```
{
   "annovar": "~/howard/databases/annovar/current/"
}
```

2.1.6 snpeff

Annotation databases folders for snpEff annotation.

Format: `A folder path (without assembly)`

Default: `~/howard/databases/snpeff/current`

Examples:

Path to snpEff database folder

```
{
   "snpeff": "~/howard/databases/snpeff/current/"
}
```

2.1.7 exomiser

Annotation databases folders for Exomiser annotation.

Format: `A folder path (without assembly)`

Default: `~/howard/databases/exomiser/current`

Examples:

Path to Exomiser database folder

```
{
   "exomiser": "~/howard/databases/exomiser/current/"
}
```

2.1.8 refseq

Annotation databases folders for refSeq annotation.

Format: `A folder path (without assembly)`

Default: `~/howard/databases/refseq/current`

Examples:

Path to refSeq files folder

```
{
   "refseq": "~/howard/databases/refseq/current/"
}
```

3 tools

External tools paths that can be defined as path to a binary or a dict including the binary type (such as "bin", "jar", "perl"). External tools can be configured with docker, using 'docker' as binary type and options to define docker 'image' (mandatory), to specify 'entrypoint', 'command' and docker 'options' (e.g. folder mount '-v /path/to/folder:/path/to/folder').

Examples:

Example of a configuration for tools, with env $PATH, full path and path with type

```
{
   "tools": {
      "bcftools": "bcftools",
      "bgzip": "bgzip",
      "java": "/usr/bin/java",
      "snpeff": "~/howard/tools/snpeff/current/bin/snpEff.jar",
```

```
        "annovar": {"jar": "~/howard/tools/annovar/current/bin/table_annovar.pl"},
        "exomiser": {"jar": "~/howard/tools/exomiser/current/bin/exomiser-cli-13.2.0.jar"}
    }
}
```

Example of a configuration for bcftools with a docker image (example with howard docker image)

```
{
    "tools": {
        "bcftools": {
            "docker": {
              "image": "howard:0.12.2.0",
              "entrypoint": "bcftools",
              "options": null,
              "command": null
            }
        }
    }
}
```

Example of a configuration for splice with a docker image

```
{
    "tools": {
        "splice": {
            "docker": {
              "image": "bioinfochrustrasbourg/splice:0.2.1",
              "entrypoint": "/bin/bash",
              "options": null,
              "command": null
            }
        }
    }
}
```

3.1 bcftools

BCFTools binary (see https://samtools.github.io/bcftools/).

Default: `bcftools`

Examples:

Path to binary in $PATH env variable

```
{
    "bcftools": "bcftools"
}
```

Path to binary as a dict with binary type 'bin'

```
{
    "bcftools": {"bin": "~/howard/tools/bcftools/current/bin/bcftools"}
}
```

3.2 bgzip

BGZip binary (see https://samtools.github.io/bcftools/).

Default: `bgzip`

Examples:

Path to binary in $PATH env variable

```
{
   "bgzip": "bgzip"
}
```

Path to binary as a dict with binary type 'bin'

```
{
   "bgzip": {"bin": "~/howard/tools/htslib/current/bin/bgzip"}
}
```

3.3 java

Java binary (see https://www.java.com).

Default: `java`

Examples:

Path to binary in $PATH env variable

```
{
   "java": "java"
}
```

Path to binary as a dict with binary type 'bin'

```
{
   "java": {"bin": "/usr/bin/java"}
}
```

3.4 snpeff

snpEff binary (see https://pcingola.github.io/SnpEff/).

Default: `~/howard/tools/snpeff/current/bin/snpEff.jar`

Examples:

Path to binary as a dict without binary type

```
{
   "snpeff": "~/howard/tools/snpeff/current/bin/snpEff.jar"
}
```

Path to binary as a dict with binary type 'jar'

```
{
   "snpeff": {"jar": "~/howard/tools/snpeff/current/bin/snpEff.jar"}
}
```

3.5 annovar

ANNOVAR perl script (see https://annovar.openbioinformatics.org/).

Default: `~/howard/tools/annovar/current/bin/table_annovar.pl`

Examples:

Path to binary as a dict without binary type

```
{
   "annovar": "~/howard/tools/annovar/current/bin/table_annovar.pl"
}
```

Path to binary as a dict with binary type 'perl'

```
{
   "annovar": {"jar": "~/howard/tools/annovar/current/bin/table_annovar.pl"}
}
```

3.6 exomiser

Exomiser binary (see https://www.sanger.ac.uk/tool/exomiser/).

Default: `~/howard/tools/exomiser/current/bin/exomiser-cli-13.2.0.jar`

Examples:

Path to binary as a dict without binary type

```
{
   "snpeff": "~/howard/tools/exomiser/current/bin/exomiser-cli-13.2.0.jar"
}
```

Path to binary as a dict with binary type 'jar'

```
{
   "snpeff": {"jar": "~/howard/tools/exomiser/current/bin/exomiser-cli-13.2.0.jar"}
}
```

3.7 splice

Splice Docker image binary (see https://hub.docker.com/r/bioinfochrustrasbourg/splice).

Default: `None`

Examples:

Configuration of Docker image

```
{
   "splice": {
      "docker": {
        "image": "bioinfochrustrasbourg/splice:0.2.1",
        "entrypoint": "/bin/bash",
        "options": null,
        "command": null
      }
   }
}
```

4 threads

Specify the number of threads to use for processing HOWARD. It determines the level of parallelism, either on python scripts, duckdb engine and external tools. It and can help speed up the process/tool. Use -1 to use all available CPU/cores. Either non valid value is 1 CPU/core.

Type: `int`

Default: `-1`

Examples:

Automatically detect all available CPU/cores

```
{
   "threads": -1
}
```

Define 8 CPU/cores

```
{
   "threads": 8
}
```

5 memory

Specify the memory to use in format FLOAT[kMG] (e.g. '8G', '12.42G', '1024M'). It determines the amount of memory for duckDB engine and external tools (especially for JAR programs). It can help to prevent 'out of memory' failures. By default (None) is 80%% of RAM (for duckDB).

Type: `str`

Format: `FLOAT[kMG]`

Default: `None`

Examples:

Automatically detect all available CPU/cores

```
{
   "threads": -1
}
```

Define 8 CPU/cores

```
{
   "threads": 8
}
```

6 assembly

Genome Assembly (e.g. 'hg19', 'hg38').

Type: `str`

Default: `hg19`

Examples:

Default assembly for all analysis tools

```
{
   "assembly": "hg19"
}
```

List of assemblies for databases download tool

```
{
   "assembly": "hg19,hg38"
}
```

7 verbosity

Verbosity level Available: CRITICAL, ERROR, WARNING, INFO, DEBUG or NOTSET

- DEBUG: Detailed information, typically of interest only when diagnosing problems.
- INFO: Confirmation that things are working as expected.
- WARNING: An indication that something unexpected happened.
- ERROR: Due to a more serious problem.
- CRITICAL: A serious error.
- NOTSET: All messages.

Type: `str`

Choices: `['CRITICAL', 'ERROR', 'WARNING', 'INFO', 'DEBUG', 'NOTSET']`

Default: `INFO`

Examples:

Default verbosity

```
{
   "verbosity": "INFO"
}
```

ERROR level (quiet mode)

```
{
   "verbosity": "ERROR"
}
```

For debug

```
{
   "verbosity": "DEBUG"
}
```

8 tmp

Temporary folder (e.g. '/tmp'). By default, '.tmp' for duckDB (see doc),external tools and python scripts.

Type: `Path`

Default: `None`

Examples:

System temporary folder

```
{
   "tmp": "/tmp"
}
```

HOWARD work directory

```
{
   "tmp": "~/howard/tmp"
}
```

Current work directory

```
{
   "tmp": ".tmp"
}
```

9 access

Access mode to variants file or database. Either 'RW' for Read and Write, or 'RO' for Read Only.

Type: `str`

Choices: `['RW', 'RO']`

Default: `RW`

Examples:

Read and Write mode

```
{
   "access": "RW"
}
```

Read only mode

```
{
   "access": "RO"
}
```

10 duckdb_settings

DuckDB settings (see duckDB doc) as JSON (string or file). These settings have priority (see options 'threads', 'tmp'...). Examples: '{"TimeZone": "GMT", "temp_directory": "/tmp/duckdb", "threads": 8}'.

Type: `Path`

Default: `None`

Examples:

DuckDB settings JSON file

```
{
   "duckdb_settings": "/path/to/duckdb_config.json"
}
```

JSON string for Time zone, temporary directory and threads for duckDB

```
{
   "duckdb_settings": {
      "TimeZone": "GMT",
      "temp_directory": "/tmp/duckdb",
      "threads": 8
   }
}
```

11 chunk_size

Number of records in batch to export output file. The lower the chunk size, the less memory consumption. For Parquet partitioning, files size will depend on the chunk size.

Type: `int`

Default: `1000000`

Examples:

Chunk size of 1.000.000 by default

```
{
   "chunk_size": 1000000
}
```

Smaller chunk size to reduce Parquet file size and memory usage

```
{
   "chunk_size": 100000
}
```

12 log

Logs file (e.g. 'my.log').

Type: `Path`

Default: `None`

Examples:

Relative path to log file

```
{
   "log": "my.log"
}
```

HOWARD work directory

```
{
   "log": "~/howard/log"
}
```

Full path to log file

```
{
   "log": "/tmp/my.log"
}
```