

HOWARD Help Configuration Prioritization

Contents

1	Introduction	1
2	Prioritization fields	3
2.1	Prioritization fields	3
2.2	Filter criteria	3
2.2.1	type	3
2.2.2	value	3
2.2.3	sql	3
2.2.4	fields	3
2.3	Filter result	3
2.3.1	score	3
2.3.2	flag	4
2.3.3	class	4
2.3.4	comment	4
2.4	Other sections	4

1 Introduction

Prioritization algorithm uses profiles to flag variants (as passed or filtered), calculate a prioritization score, classify with keywords, and automatically generate a comment for each variants (example: 'polymorphism identified in dbSNP. associated to Lung Cancer. Found in ClinVar database'). Prioritization profiles are defined in a configuration file in JSON format. A profile is defined as a list of filters, using SQL syntax or wildcards and comparison options (contains, lower than, greater than, equal...). Filters uses all annotations fields within VCF INFO/Tags, provided by annotations tools, such as HOWARD itself (example: COSMIC, Clinvar, 1000genomes, PolyPhen, SIFT).

See HOWARD Help Prioritization tool tool for more information.

This example describes the prioritization profile 'default' that uses 2 fields 'DP' ('Read Depth') and 'CLNSIG' ('ClinVar Significance') with specific criteria.

The 'DP' filter is related to 'DP' field, 2 filters are applied: if 'DP' is greater than or equal to '50', score is '5' and flag is 'PASS', if 'DP' is lower than '50', score is '0' and flag is 'FILTERED'. It means that if 'Read depth' is lower than 50 for a variant, it will be filtered with a bad score. Otherwise, the variant will have a better score (but can be filtered because of other filters).

The 'CLNSIG' filter is related to field 'CLNSIG', 2 filters are applied: if it is equal to 'pathogenic', score is '15' and flag is 'PASS', and if it is equal to 'non-pathogenic', score is '-100' and flag is 'FILTERED'. Thus, the variant will be well scored if it is pathogenic, but filtered with a bad score if it is non pathogenic (for Clinvar annotation).

The 'Class' filter combines 2 fields ('DP' and 'CLNSIG') in SQL syntax, within 2 different filters: 1. filter 1: associated with a score of '100', a lflag of 'PASS' and classifications as 'PM1' and 'PM2' 2. filter 1: associated with a score of '200', a lflag of 'PASS' and classifications as 'PM1' and 'PM3'

Examples:

Example of a configuration JSON file

```
{
  "default": {
    "_description": "Default prioritization profile",
```

```

    "_version": "1.0.0",
    "DP": [
      {
        "type": "gte",
        "value": "50",
        "fields": ["DP"],
        "score": 5,
        "flag": "PASS",
        "comment": [
          "DP higher than 50"
        ]
      },
      {
        "type": "lt",
        "value": "50",
        "fields": ["DP"],
        "score": 0,
        "flag": "FILTERED",
        "comment": [
          "DP lower than 50"
        ]
      }
    ],
    "CLNSIG": [
      {
        "type": "equals",
        "value": "pathogenic",
        "fields": ["CLNSIG"],
        "score": 15,
        "flag": "PASS",
        "comment": [
          "Described on CLINVAR database as pathogenic"
        ]
      },
      {
        "type": "equals",
        "value": "non-pathogenic",
        "fields": ["CLNSIG"],
        "score": -100,
        "flag": "FILTERED",
        "comment": [
          "Described on CLINVAR database as non-pathogenic"
        ]
      }
    ],
    "Class": [
      {
        "sql": " DP >= 100 OR regexp_matches(CLNSIG, 'Pathogenic') ",
        "fields": ["DP", "CLNSIG"],
        "score": 100,
        "flag": "PASS",
        "class": "PM1,PM2",
        "comment": [
          "Described on CLINVAR database as pathogenic, classified as PM1 and PM2"
        ]
      },
      {
        "sql": ["DP >= 200", "OR regexp_matches(CLNSIG, 'Pathogenic')"],
        "fields": ["DP", "CLNSIG"],

```

```

    "score": 200,
    "flag": "PASS",
    "class": ["PM1", "PM2"],
    "comment": [
        "Described on CLINVAR database as non-pathogenic, classified as PM1 and PM3"
    ]
  }
]
}
}

```

2 Prioritization fields

2.1 Prioritization fields

A prioritization profile contains filters applied to specific fields to prioritize or filter variants based on defined criteria. Each profile is identified by a unique name (e.g. 'default') , and defines filters applied to a specific field (e.g. 'DP', 'CLNSIG'). Each filter consists of criteria such as type of test (e.g. greater than, contains), threshold value or substring, and return a score, a flag, and a comment if the test is valid.

2.2 Filter criteria

2.2.1 type

- *Type*: String
- *Description*: Specifies the type of test to apply to the field (INFO/Tags). It can be
 - 'gt' for 'greater than' ('>')
 - 'gte' for 'greater than or equal to' ('>=')
 - 'lt' for 'lower than' ('<')
 - 'lte' for 'lower than or equal to' ('<=')
 - 'equals' for exact match ('='), for integer or string comparison
 - 'contains' for exact match

2.2.2 value

- *Type*: String or Integer
- *Description*: Specifies the threshold value or string/substring to match within the field, depending on the type of test specified.

2.2.3 sql

- *Type*: String (or Array of String) with SQL syntax
- *Description*: Specifies the filter on fields (INFO/Tags) in SQL syntax (beware of fields type and cast if necessary, and beware of null values).

2.2.4 fields

- *Type*: Array of String
- *Description*: Specifies the fields (INFO/Tags) used in operation or SQL syntax.

2.3 Filter result

2.3.1 score

- *Type*: - *Type*: Integer

- *Description:* Assigns a score to variants that pass the filter. The score is used for prioritization or ranking purposes. Depending on the prioritization calculation mode, this score will be incremented ('HOWARD' mode) or compared if it is the max ('VaRank' mode).

2.3.2 flag

- *Type:* String
- *Description:* Assigns a flag (either 'PASS' or 'FILTERED') to variants that pass the filter. The flag provides additional information or categorization about the variant. Flag 'FILTERED' is prior to 'PASS' to calculate the global prioritization profile flag.

2.3.3 class

- *Type:* Array of String
- *Description:* Assigns multiple classes (e.g. 'PM1', 'PM2', 'ClassA') to variants that pass the filter.

2.3.4 comment

- *Type:* Array of String
- *Description:* Provides a comment or explanation for the filter criteria. It helps users understand the rationale behind applying the filter.

2.4 Other sections

More sections can be added for information on profiles by using '_' as first character (such as '__description', '__version')