

HOWARD Help Configuration Calculation

Contents

1	Introduction	1
2	Calculation fields	2
2.1	type	2
2.2	name	2
2.3	description	2
2.4	available	2
2.5	output_column_name	2
2.6	output_column_type	2
2.7	output_column_description	2
2.8	operation_query	3
2.9	info_fields	3
2.10	operation_info	3
2.11	function_name	3
2.12	function_params	3

1 Introduction

Calculation processes variants annotations to generate new annotation, such as: identify variation type (VarType), harmonizes allele frequency (VAF) and calculate statistics (VAF_stats), extracts Nomen (transcript, cNomen, pNomen...) from an HGVS field (e.g. snpEff, Annovar) with an optional list of personalized transcripts, generates VaRank format barcode, identify trio inheritance. These calculations are based on existing annotations of variants (and genotypes).

See HOWARD Help Calculation tool tool for more information.

Calculations are either provided by HOWARD within code, or configured into a JSON file with specific fields to define how to calculate the new annotation. Calculations are either an inner HOWARD Python code, or a SQL query.

Examples:

Example of a configuration JSON file

```
{
  "VARTYPE": {
    "type": "sql",
    "name": "VARTYPE",
    "description": "Variant type (e.g. SNV, INDEL, MNV, BND...)",
    "available": true,
    "output_column_name": "VARTYPE",
    "output_column_type": "String",
    "output_column_description": "Variant type: SNV if X>Y, MOSAIC if X>Y,Z or X,Y>Z, INDEL if XY>Z or X
    "operation_query": [
      "CASE",
      "WHEN \"SVTYPE\" NOT NULL THEN \"SVTYPE\"",
      "WHEN LENGTH(REF) = 1 AND LENGTH(ALT) = 1 THEN 'SNV'",
      "WHEN REF LIKE '%,%' OR ALT LIKE '%,%' THEN 'MOSAIC'",
      "WHEN LENGTH(REF) == LENGTH(ALT) AND LENGTH(REF) 1 THEN 'MNV'",
      "WHEN LENGTH(REF) <LENGTH(ALT) THEN 'INDEL'",
      "ELSE 'UNDEFINED'",
    ]
  }
}
```

```

        "END"
    ],
    "info_fields": ["SVTYPE"],
    "operation_info": true
},
"variant_id": {
    "type": "python",
    "name": "variant_id",
    "description": "Variant ID generated from variant position and type",
    "available": true,
    "function_name": "calculation_variant_id",
    "function_params": []
}
}

```

2 Calculation fields

2.1 type

- *Type:* String
- *Description:* Indicates the type of calculation. It can be either 'sql' for a SQL query or 'function' for a Python function.

2.2 name

- *Type:* String
- *Description:* Represents the unique name assigned to the calculation.

2.3 description

- *Type:* String
- *Description:* Provides a brief description of the calculation, explaining its purpose or what it calculates.

2.4 available

- *Type:* Boolean
- *Description:* Specifies whether the calculation is available and enabled for use. If set to true, it's available; otherwise, it's not accessible.

2.5 output_column_name

- *Type:* String
- *Description:* Specifies the output column and VCF INFO/Tag generated by the calculation.

2.6 output_column_type

- *Type:* String
- *Description:* Specifies the VCF INFO/Tag type of the output column generated by the calculation (e.g., String, Integer, Float).

2.7 output_column_description

- *Type:* String
- *Description:* Provides a detailed description of the the VCF INFO/Tag, explaining its content, format, or how it's derived.

2.8 operation_query

- *Type:* Array of String
- *Description:* For SQL type calculations, this field contains the SQL query used to perform the calculation. It consists of SQL statements that define the logic of the calculation. SQL columns need to be available for the query (see 'info_fields').

2.9 info_fields

- *Type:* Array of String
- *Description:* Specifies additional input fields or columns required by the calculation. These fields are used as parameters or conditions in the calculation process (especially SQL query).

2.10 operation_info

- *Type:* String
- *Description:* Indicates whether additional information (specified in 'info_fields') is necessary for performing the calculation. If set to true, the additional information is required, the INFO/tag need to be in the VCF header; otherwise, it's optional.

2.11 function_name

- *Type:* String
- *Description:* For Python type calculations, this field specifies the name of the Python function used to perform the calculation.

2.12 function_params

- *Type:* Array of String
- *Description:* For Python type calculations, specifies any additional parameters required by the Python function specified in 'function_name'.