LandersLabUtilities

By Landers Laboratory UMASS Chan Medical School
Supervisor: John Landers
Mantainers: Alberto Brusati, Pamela Keagle

Introduction

This documentation provides details about the functions defined in the provided python package (landerslabutilities). These functions, available as a python package on PYPI, were built to enlight the burden of filtering data from the ALS Compute project. In particular, we included gene table extraction, single sample analysis, joint VCF creation, and region extraction. Examples for each function are provided below.

Installation

Landerslabutilities are completely available on pypi repository. To easily access to the functionality, simply install the package via pip:

>>> ***pip3 install landerslabutils***

And import with the statement:

>>> ***import landerslabutils as llu***

Functions

1. ***genetable_fromMT(****db=None, phenoFile=None, samples=None, gene=None, variant_type=None, output****)***

   Description: Filters a Hail MatrixTable based on gene symbol and consequence type, then exports the results to a TSV file.

   Parameters:
   - db (str, required): Path to the Hail database (MT) file.
   - phenoFile (str, required): Path to the phenotype file.
   - samples (str/list, optional): Subsample your table based on a single sample or list.
   - gene (str/list, required): Gene symbol(s) to filter on.
   - variant_type (str/list, optional): Consequence type(s) to filter on.
   - output (str, required): Path to export the filtered table.

Example of usage:

```
>>> llu.genetable_fromMT(db= ’/path/to/hail.mt',
                         phenoFile='/path/to/phenotype_table.tsv ',
                         gene="SOD1",
                         variant_type=["missense_variant”,"stop_gained"],
                         output="/path/to/output_table.tsv")

>>> llu.genetable_fromMT(db= ’/path/to/hail.mt',
                         phenoFile='/path/to/phenotype_table.tsv ',
                         gene=["SOD1","TARDBP"],
                         variant_type=["missense_variant","stop_gained"],
                         output="/path/to/output_table.tsv ")
```

2. ***singlesample_fromMT****(db=None, phenoFile=None, sample, gene=None, variant_type=None, output_type="table", output_path, output_name)*

   Description: Filters a Hail MatrixTable based on a single sample and optionally gene and variant type, then exports the results.

   Parameters:

   - db (str, required): Path to the Hail database (MT) file.
   - phenoFile (st, requiredr): Path to the phenotype file.
   - sample (str, required): Sample ID to filter on.
   - gene (str/list, optional): Gene symbol(s) to filter on.
   - variant_type (str/list, optional): Consequence type(s) to filter on.
   - output_type (str, required): "table" or "vcf". Defaults to "table".
   - output_path (str, required): Path to export the filtered results.
   - output_name (str, required): Name of the output file.

   Example of usage:

```
>>> llu. singlesample_fromMT(db= ’/path/to/hail.mt',
                             phenoFile='/path/to/phenotype_table.tsv ',
                             sample="sample",
                             gene=["TARDBP", "SOD1", "FUS],
                             variant_type=None,
                             output_type = "table",
                             output_path="/path/to/lacation/",
                              output_name="sample.table.tsv")

>>> llu. singlesample_fromMT(db= ’/path/to/hail.mt',
                             phenoFile='/path/to/phenotype_table.tsv ',
                             sample="sample",
                             output_type = "vcf",
                             output_path="/path/to/lacation/",
                              output_name="sample.table.vcf.gz")
```

3. ***jointVCF_fromMT****(db=None, samples_file=None, output_path=None, output_name=None)*

   Description: Filters a Hail MatrixTable and creates joint VCF files for each chromosome.

   Parameters:

   - db (str, required): Path to the Hail database (MT) file.
   - samples_file (str, required): Path to the subsample file.
   - output_path (str, required): Path to export the VCF files.
   - output_name (str, required): Name of the output files.

   Example of usage:

```
>>> llu.jointVCF_fromMT(db=’/path/to/hail.mt',
                        phenoFile='/path/to/phenotype_table.tsv ',
                        samples_file="/path/to/subsample.txt",
                        output_path="/path/to/lacation/",
                        output_name="subsample.vcf.gz")
```

4. ***extract_regions****(db=None, phenoFile=None, samples=None, gene=None, region=None, output_type="table", output=None)*

   Description: Filters a Hail MatrixTable based on genomic regions and optionally gene and variant type, then exports the results.

   Parameters:

   - db (str): Path to the Hail database (MT) file.
   - phenoFile (str): Path to the phenotype file.
   - samples (str): Sample ID(s) to filter on.
   - gene (str/list, optional): Gene symbol(s) to filter on.
   - region (list): Regions to extract.
   - output_type (str): "table" or "vcf". Defaults to "table".
   - output (str): Path to export the filtered results.

   Example of usage:

```
>>> llu.extract_regions(db=’/path/to/hail.mt',
                        phenoFile='/path/to/phenotype_table.tsv ',
                        samples="sample_id",
                        region= ['chr13:32355000-32375000'],
                        output_type="table",
                        output_name="subsample.vcf.gz")
```