

Computing for Medicine

Project 6: Disease data capture using web technologies

1 PhenoTips and the Human Phenotype Ontology (HPO)

The PhenoTips web application [1] is used to collect and analyze patient information, including phenotypic information: <https://phenotips.org/>. PhenoTips is an open source software tool for collecting and analysing phenotypic information for patients with genetic disorders. The user interface closely mirrors clinician workflows so as to facilitate the recording of observations made during the patient encounter. This easy-to-use front-end, compatible with any device that runs a Web browser, is coupled with a standardized database back-end.

PhenoTips uses the Human Phenotype Ontology (HPO) [2] to name and classify phenotypic information: <http://human-phenotype-ontology.github.io/>. The HPO is a hierarchical representation of phenotypes (signs and symptoms, especially relevant for genetic diseases). These phenotypes are organized using an is-a relationship, from very general ("Phenotypic abnormality") to more specific ("abnormality of the hand") to very specific ("2-3 finger cutaneous syndactyly"). Overall the HPO contains > 12,000 phenotypic terms, most of which are linked to relevant genetic disorders (OMIM database).

2 Part 0: Explore PhenoTips

To begin, visit the PhenoTips playground (<https://playground.phenotips.org/>) and use the PhenoTips tool. This instance of PhenoTips is publicly accessible and can be used without registering an account. Play with PhenoTips by creating new patient records and searching for existing patient records. As you examine the records, pay special attention to the phenotypic information recorded for each patient.

3 Part 1: Use the PhentoTips API

In your first task for this project, you interacted with the PhenoTips website using a web browser. Now, you will perform some of the same tasks, and others, using a Python program.

3.1 PhenoTips API

The developers of the PhenoTips web application have provided an API, which allows software developers to write programs that interact with PhenoTips. Documentation and examples of using the PhenoTips API are posted here:

<https://phenotips.org/DevGuide/RESTfulAPI#HBasicexamplesusingaRESTclient>

Using the PhenoTips API, we can write code to retrieve a particular patient record or all patient records from an instance of PhenoTips. We can also make changes to patient records, add new patient records, and even delete patient records.

The data retrieved from PhenoTips is returned in a JSON format that is described here:

<https://phenotips.org/DevGuide/JSONExport>

3.2 Your Tasks

In `phenotips_project.py`, there are three complete functions. Read those functions, along with the PhenoTips API documentation, to gain a better understanding of how to write code that interacts with PhenoTips. Next, complete the bodies of the following three functions:

- `get_patients_range`
- `delete_patient`
- `get_phenotypic_info`

To test your code, use the example function calls in the main block. You can also use the PhenoTips website search functionality to check whether your code performed as expected.

4 Part 2: Human Phenotype Ontology (HPO)

The Human Phenotype Ontology (<http://human-phenotype-ontology.github.io/>) is a classification of human phenotypic abnormalities observed in human disease. The HPO has a hierarchical organization with four main subcategories: *mortality/aging*, *phenotypic abnormality*, *clinical modifier*, and *mode of inheritance*. Figure 1 shows this hierarchy, with the name of each category and its unique identifier. In this project, you will work only with the subcategory *phenotypic abnormality*.

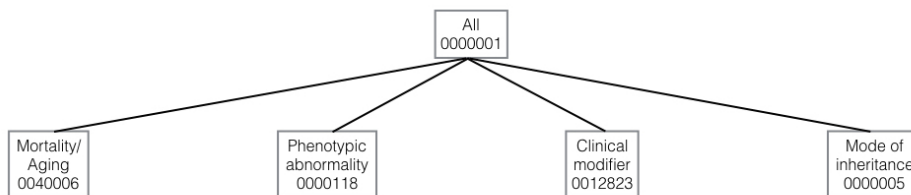


Figure 1: HPO’s four subcategories

Figure 2 shows part of the HPO *phenotypic abnormality* ontology. Each phenotypic abnormality has a name and a unique numeric identifier. The root of the hierarchy has ID 0000118 and 23 subcategories, including the three that are shown: Abnormality of the ear (0040006), Abnormality of the breast (0000769), and Abnormality of the voice (0001608). Each of those phenotypic features may have subcategories as well, those subcategories may have subcategories, and so on.

For each phenotypic abnormality, we will use a level number to describe its distance from the root. For example, Abnormality of the voice, is at level 1, Hyponasal speech is at level 2, and Laryngeal dystonia is at level 3.

As part of the project files, we included the HPO file, `hp.obo` (accessed from the HPO website on 11 January 2017). We also provided a Python program `ontology_parser.py`, which reads and parses the HPO file, and produces two dictionaries containing information about the ontology:

- `pid_to_name`: each key is an ID for a phenotypic feature and each value is its name
- `pid_to_parents`: each key is an ID for a phenotypic feature and each value is a list of its parent IDs

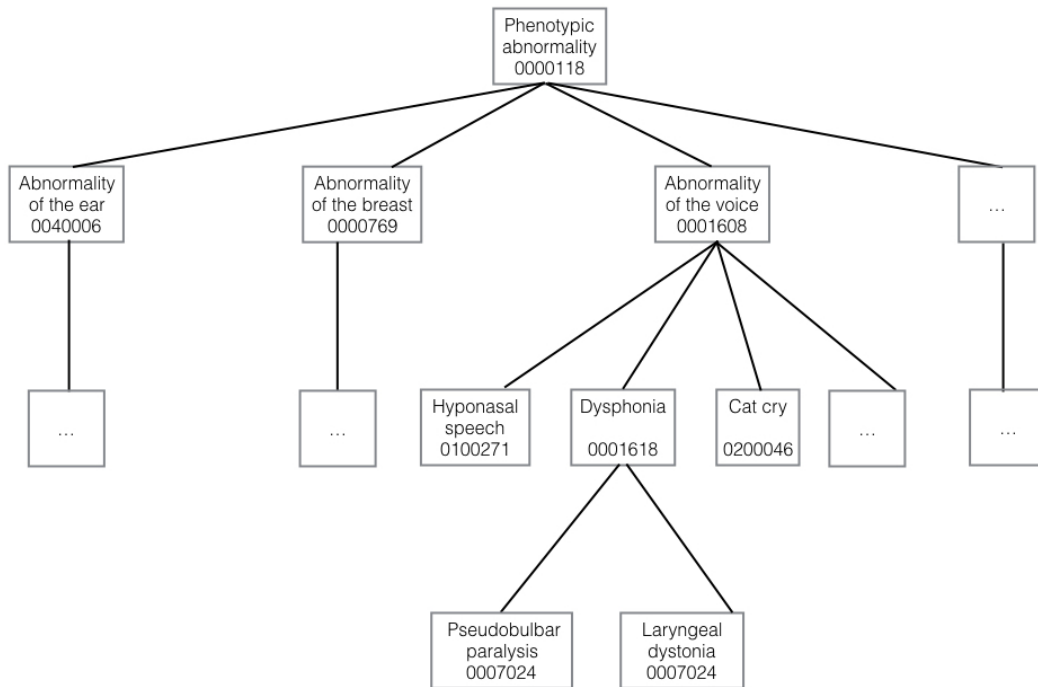


Figure 2: Part of the phenotypic abnormality ontology

4.1 Your tasks

Starter code file `ontology_explorer.py` contains several incomplete functions. Your task is to complete these functions:

- `get_single_path`
- `get_ids_at_level`
- `get_all_paths`
- `get_all_paths_to_level`

Here are a few tips:

- Spend time thinking about the algorithms, before starting to write code.
- Trace examples on paper and use the debugger.
- If your functions get long or can be divided into parts, consider writing helper functions to simplify your code.
- As you implement each function, call on the examples from the main block to test your work. Add example calls of your as well.
- `get_all_paths` and `get_all_paths_to_level` are very similar. Focus on getting one of those two functions working, before working on the other.

5 Part 3: Q & A

In a file named `project6.txt`, answer the questions below. For some questions, you may find it helpful to write some code or call on your existing functions to find the answer. For others, no programming is needed.

1. Instead of using `get_all_paths`, we can use `get_all_paths_to_level` to find all paths to the root. Describe which arguments to pass to `get_all_paths_to_level` to achieve this.
2. In HPO, what is the maximum number of parents that any phenotypic feature has? Give the name and ID of one phenotypic feature that has the maximum number of parents.
3. In HPO, what is the maximum level (distance from the root) of any feature in the ontology? Give the name and ID of one phenotypic feature at that level.

References

- [1] Marta Girdea, Sergiu Dumitriu, Marc Fiume, Sarah Bowdin, Kym M Boycott, Sébastien Chénier, David Chitayat, Hanna Faghfoury, M Stephen Meyn, Peter N Ray, et al. Phenotips: patient phenotyping software for clinical and research use. *Human mutation*, 34(8):1057–1065, 2013.
- [2] Sebastian Köhler, Nicole A Vasilevsky, Mark Engelstad, Erin Foster, Julie McMurry, Ségolène Aymé, Gareth Baynam, Susan M Bello, Cornelius F Boerkoel, Kym M Boycott, et al. The human phenotype ontology in 2017. *Nucleic Acids Research*, page gkw1039, 2016.