Processing phenotype data using Phenopackets-API and PXFTools

Christopher J Mungall[1*], Jules Jacobsen[2*], James Balhoff[3], Jeremy Nguyen-Xuan[1], Kent Shefchek[4], Dan Keith[4], Harry Hochheiser[5], Suzanna E Lewis[1], Sebastian Köhler[6], Peter Robinson[6], Julie McMurray[4], Tudor Groza[7], Melissa Haendel[4]

[1] Lawrence Berkeley National Laboratory, Berkeley, CA, USA. Email: cjmungall@lbl.gov

[2] Sanger Institute, Hinxton, UK.

[3] RTI International, Durham, NC, USA.

[4] Oregon Health and Sciences University, Portland, OR, USA.

[5] University of Pittsburgh, Pittsburgh, PA, USA.

[6] Charité – Universitätsmedizin Berlin, Germany.

[7] Garvan Institute of Medical Research, Sydney, Australia

[*]These authors contributed equally.

Project Website: http://phenopackets.org
Source Code: https://github.com/phenopackets

While great strides have been made in exchange formats for genomic sequence and variation data (e.g. Variant Call Format; VCF), the same is not true for phenotypic features. Similarly, bioinformatics software libraries such as BioPython and BioPerl have rich object models for genomic or phylogenetic datatypes, but lack a uniform phenotype representation. This is due in part to the diversity of phenotypic descriptions, from clinical observations through QTLs and newer high-throughput phenopype measurements. As a result, phenotypes are represented differently in different databases, making it harder to exchange, aggregate and operate over phenotypic data.

We have designed a datamodel and exchange format standard for flexible, extensible and expressive representation of a broad range of phenotypes in humans or any other species. The Phenotype eXchange Format (PXF) works hand-in-hand with phenotype ontology such as the Human Phenotype Ontology or the Ontology of Biological Attributes, but allows for representation of other fields such as quantitative measurements, environments and evidence. PXF can be serialized as either JSON or YAML, and we provide JSON schema for validation. We also provide a JSON-LD context for use within semantic web and OWL stacks.

For software developers we provide the Phenopacket Application Programmer Interfaces (APIs) in Java, Python and Javascript. We have implemented bindings for Neo4J via SciGraph (https://github.com/SciGraph/SciGraph), but the model is storage-layer independent and can potentially be used in conjunction with GMOD databases such as Chado. We are also prototyping a language-independent ProtoBuf-API to facilitate interoperability with the Global Alliance for Genomes and Health (GA4GH) stack.

As a demonstration of purpose, we have implemented a command line tool library called pxftools, analogous to the popular vcftools used for operating over variant files.