INTERNATIONAL STANDARD

ISO 4454

First edition
2022-07

Genomics informatics — Phenopackets: A format for phenotypic data exchange

Reference number
ISO 4454:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 215, *Health informatics*, Subcommittee SC 1, *Genomics informatics*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

While great strides have been made in exchange formats for sequence and variation data (e.g. Variant Call Format), the majority of genotype formats do not include a means to share corresponding phenotypic (e.g. observable characteristics, signs/symptoms of disease) information. While some genomic databases have defined their own formats for representing phenotypic information, the lack of uniformity amongst these organizations hinders communication and limits the ability to perform analysis across organizations. For individuals with rare and undiagnosed disease, broad adoption and utilization of uniform, machine-readable, phenotypic descriptions could improve the speed and accuracy of diagnosis by promoting quicker, more comprehensive and cost-effective information acquisition and exchange relevant for research and medical care.

Phenotypic abnormalities of individuals are currently described in diverse places in diverse formats, such as journal/publications databases, laboratory systems, patient registries, health records, and even in social media. The structure of the data in the phenopackets exchange standard will be optimized for integration and efficient data flow across these distributed contexts. Increasing the volume of computable data across a diversity of systems will support large-scale computational disease analysis of combined genotype and phenotype data. Studies of well over 100 000 patients are thought to be required to effectively assess the role of rare variation in common disease or to discover the genomic basis for a substantial portion of diseases. Phenopackets can help integrate geographically distributed cases to build such virtual cohorts and remove the time burden on resources that need to integrate information manually.

Medical coding systems and clinical exchange standards have not to date included rich phenotypic descriptions, as they are largely focused on supporting billing and clinical encounter documentation, rather than the documenting and sharing of the biologically relevant phenotypic information needed for computational use, mechanism discovery, and precision classification. From a clinical perspective, the integration of a standard for phenotypic description and exchange into and out of EHRs would improve disease diagnosis and management, especially for genomic health and precision medicine applications.

Phenopackets enable clinicians, biologists, and disease and drug researchers to build more complete models of disease. It is designed to encourage wide adoption and synergy between the people, organizations and systems that comprise the joint effort to address human disease and biological understanding. The phenopacket proposed in this document is designed to support deep phenotyping, a process wherein individual components of each phenotype are observed and documented. The phenotypes can be constitutional or those related to a sample (such as from a biopsy).

Genomics informatics — Phenopackets: A format for phenotypic data exchange

1 Scope

This document specifies a uniform, machine-readable, phenotypic description of an individual, patient or sample in the context of rare disease, common/complex disease or cancer.

It is applicable to academic, clinical and commercial research, as well as clinical diagnostics. While intended for human data collection, it can be used in other areas (e.g. mouse research). It does not define the phenotypic information that needs to be collected for a particular use but represents that information in an appropriately descriptive manner that allows it to be computationally exchanged between systems.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601 (all parts), *Date and time — Representations for information interchange*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
biosample
unit of biological material from which the substrate for analysis is extracted to support the assessment, diagnosis, treatment, mitigation or prevention of a disease, disorder, abnormal physical state or its symptoms

3.2
boolean
data type having two values: one and zero (which are equivalent to true and false)

[SOURCE: ISO 2146:2010, 4.6.1]

3.3
CURIE
compact URI
generic, abbreviated syntax for expressing *uniform resource identifiers* (3.22)

3.4
deletion
variation in which a part of a chromosome or sequence of DNA is lost relative to a *reference sequence* (3.17)

3.5
DNA sequence
order of *nucleotide bases* (3.10) (adenine, guanine, cytosine and thymine) in a molecule of DNA

3.6
exome sequencing
technique for sequencing the protein-coding genes in a genome

[SOURCE: ISO/TS 20428:2017, 3.38, modified — "whole" removed from preferred term.]

3.7
gene
basic unit of hereditary information composed of chains of nucleotides in specific sequences that encodes a protein or protein subunit

[SOURCE: ISO 11238:2018, 3.29]

3.8
gestational age
menstrual age
time elapsed between the first day of the last normal menstrual period and the day of delivery

Note 1 to entry: The first day of the last menstrual period occurs approximately 2 weeks before ovulation and approximately 3 weeks before implantation of the blastocyst. Because most women know when their last period began but not when ovulation occurred, this definition traditionally has been used when estimating the expected date of delivery. In contrast, chronological age (or postnatal age) is the time elapsed after birth.

3.9
insertion
addition of one or more *nucleotide base pairs* (3.10) into a *DNA sequence* (3.5)

[SOURCE: ISO/TS 20428:2017, 3.19]

3.10
nucleotide base
nucleotide base pair
monomer of a nucleic acid polymer such as DNA or RNA

Note 1 to entry: Nucleotides are denoted as letters (‘A’ for adenine; ‘C’ for cytosine; ‘G’ for guanine; ‘T’ for thymine that only occurs in DNA; and ‘U’ for uracil that only occurs in RNA). The chemical formula for a specific DNA or RNA molecule is given by the sequence of its nucleotides, which can be represented as a string over the alphabet (‘A’, ’C’, ’G’, ‘T’) in the case of DNA, and a string over the alphabet (‘A’, ‘C’, ‘G’, ‘U’) in the case of RNA. Bases with unknown molecular composition are denoted with ‘N’.

[SOURCE: ISO 23092-2:2020, 3.20]

3.11
ontology
logical structure of the *terms* (3.20) used to describe a domain of knowledge, including both the definitions of the applicable terms and their relationships

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.2691]

3.12
pedigree
structured description of the familial relationships between samples

Note 1 to entry: Pedigree information is represented in the form of a PED file.

3.13
phenopacket
uniform, machine-readable, phenotypic description of an individual, patient or sample

Note 1 to entry: Includes a catch-all collection of data types, specifically focused on representing disease data in both initial data capture and analysis.

3.14
phenotype
set of observable characteristics of an organism resulting from the interaction of its genotype with the environment

Note 1 to entry: 'Phenotypic feature' is a descriptive feature, such as Arachnodactyly, that is the component of a disease, such as Marfan syndrome. It can be observed as either present or absent (excluded), with possible onset, modifiers and frequency.

3.15
proband
affected family member who seeks medical attention thereby bringing the family under study

3.16
quality score
quality value
number assigned to each *nucleotide base* (3.10) call in automated sequencing processes

Note 1 to entry: Quality values express the base-call accuracy, i.e. the probability (or a related measure) for a nucleotide in the sequence to have been incorrectly determined.

[SOURCE: ISO 23092-2:2020, 3.22, modified — First preferred term and admitted term have been swapped.]

3.17
reference sequence
nucleic acid sequence used either to align by mapping sequence reads or as the basis for annotations such as genes and sequence variations

[SOURCE: ISO 20397-2:2021, 3.26]

3.18
single nucleotide polymorphism
SNP
single nucleotide variation in a genetic sequence that occurs at appreciable frequency in the population

Note 1 to entry: Pronounced "snip".

Note 2 to entry: Can also be referred to as single nucleotide variation (SNV).

[SOURCE: ISO 25720:2009, 4.23, modified — Note 1 and Note 2 to entry have been added.]

3.19
string
data type consisting of a sequence of one or more characters

[SOURCE: ISO 2146:2010, 4.6.9]

3.20
term
ontology class composed of a definition, a label, and a unique identifier

3.21
UBERON
comparative anatomy ontology representing a variety of structures found in animals, such as lungs, muscles, bones, feathers and fins

3.22
uniform resource identifier
URI
string (3.19) of characters that unambiguously identifies a particular resource, such as registered name spaces or protocols

3.23
variant
alteration in the most common DNA nucleotide sequence

Note 1 to entry: It can describe an alternation that can be benign, pathogenic, or of unknown significance.

Note 2 to entry: Variant implies deletion, insertion, indel or single nucleotide polymorphism.

4 Abbreviated terms

ACMG	American College of Medical Genetics
AJCC	American Joint Committee on Cancer
CNV	Copy Number Variation
DNA	Deoxyribonucleic Acid
ECO	Evidence and Conclusion Ontology
EHR	Electronic Health Record
GENO	Genotype Ontology
HGNC	HUGO Gene Nomenclature Committee
HGVS	Human Genome Variation Society
HPO	Human Phenotype Ontology
HTS	High-Throughput Sequencing
HUGO	Human Genome Organization
ICD	International Classification of Diseases
IRI	Internationalized Resource Identifier
ISCN	International System for Human Cytogenomic Nomenclature
MONDO	Mondo Disease Ontology
NCIT	National Cancer Institute Thesaurus
OBO	Open Biological and Biomedical Ontology
OMIM	Online Mendelian Inheritance in Man
PDX-MI	Patient-derived tumor xenograft minimal information standard
PURL	Persistent Uniform Resource Locator
RNA	Ribonucleic Acid
SAM	Sequence Alignment Map

SPDI	Sequence Position Deletion Insertion
TNM	Classification of Malignant Tumors
URL	Uniform Resource Locator
VCF	Variant Call Format
VRS	Variation Representation Specification

5 Phenopackets Schema and Requirements

5.1 Phenopacket Schema

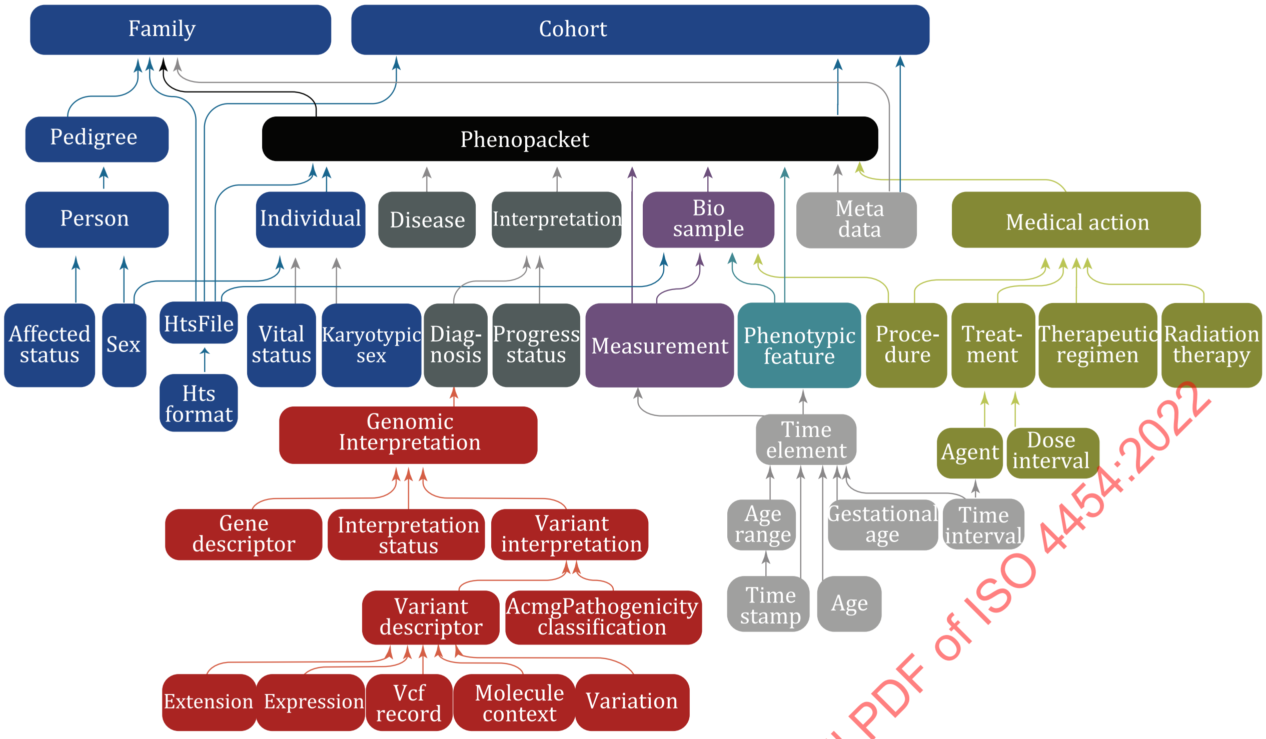
The phenopacket schema contains a common, limited set of data types which can be composed into more specialized types for data sharing between resources using an agreed upon common schema. There are three top-level elements – Phenopacket, Family, and Cohort – with other properties, or 'building blocks', nested within. An overview of schema elements and their thematic groupings can be found in Figure 1, with detailed class diagrams of those thematic groupings shown in Figure 2.

The phenopacket is formally defined in protobuf3[1]. Protobuf is language-neutral, faster than other schema languages such as XML and JSON and can be simpler to use because of features such as automatic validation of data objects. It also works with many languages, including Java, GO, C#, C++, JS and Python.[2] See Annex A for several examples that demonstrate how to work with phenopackets in Java and C++.

Given the nested nature of phenopackets elements, it can be difficult to understand the overall structure and relationships within phenopackets in a linear document. The documentation for the phenopacket-schema with hyperlinked building blocks can be found at https://phenopacket-schema.readthedocs.io/en/v2/index.html.

[1] Protobuf is an exchange format developed by Google LLC. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO.

[2] These trademarks are examples of suitable products available commercially. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO of these products.



Key

High-level themes

● core phenopacket

● biosample classes

● base classes: person/family/cohort

● disease interpretation classes

● genomic interpretation classes

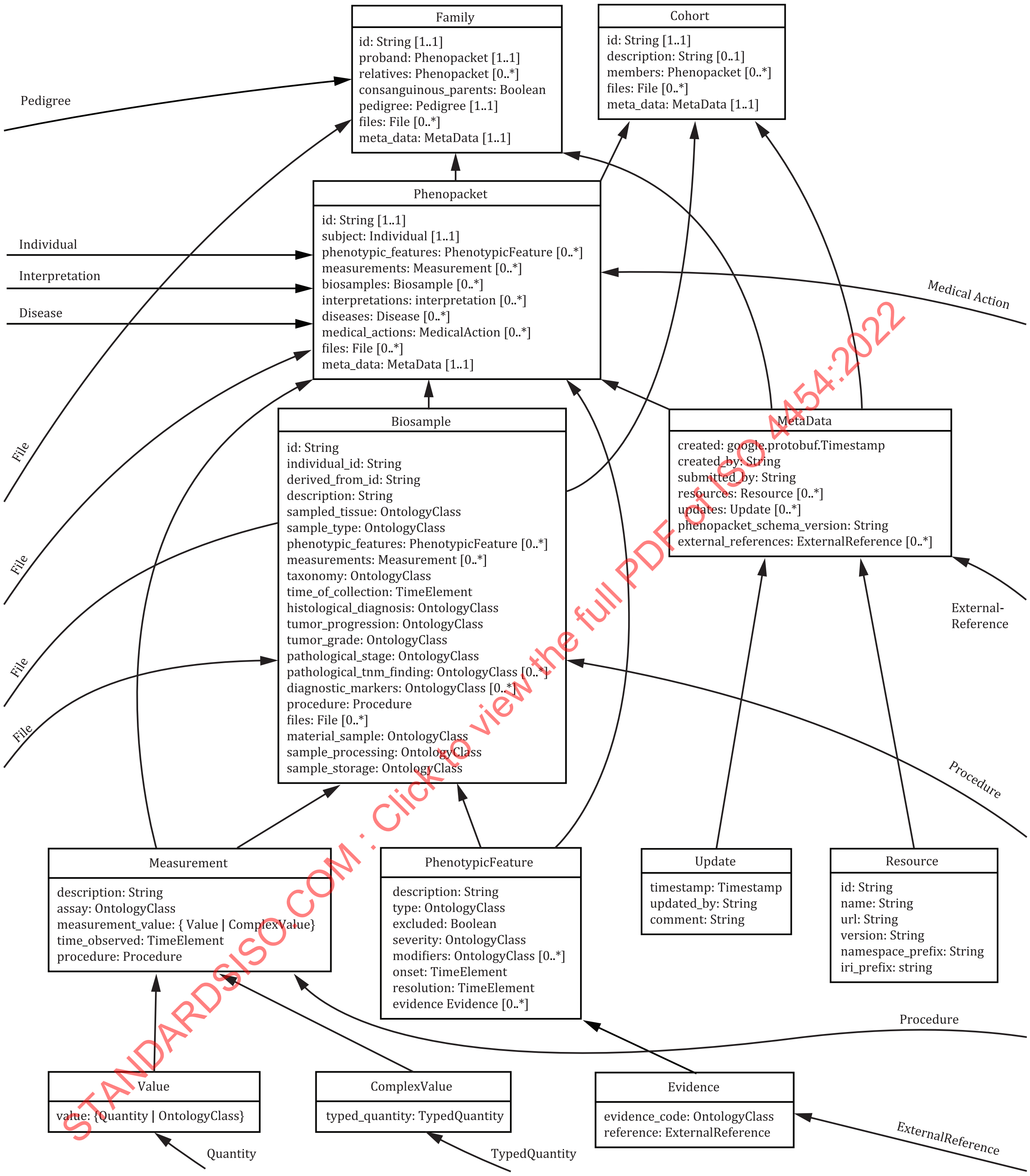
● medical action classes

● phenotypic features

● multi-purpose classes

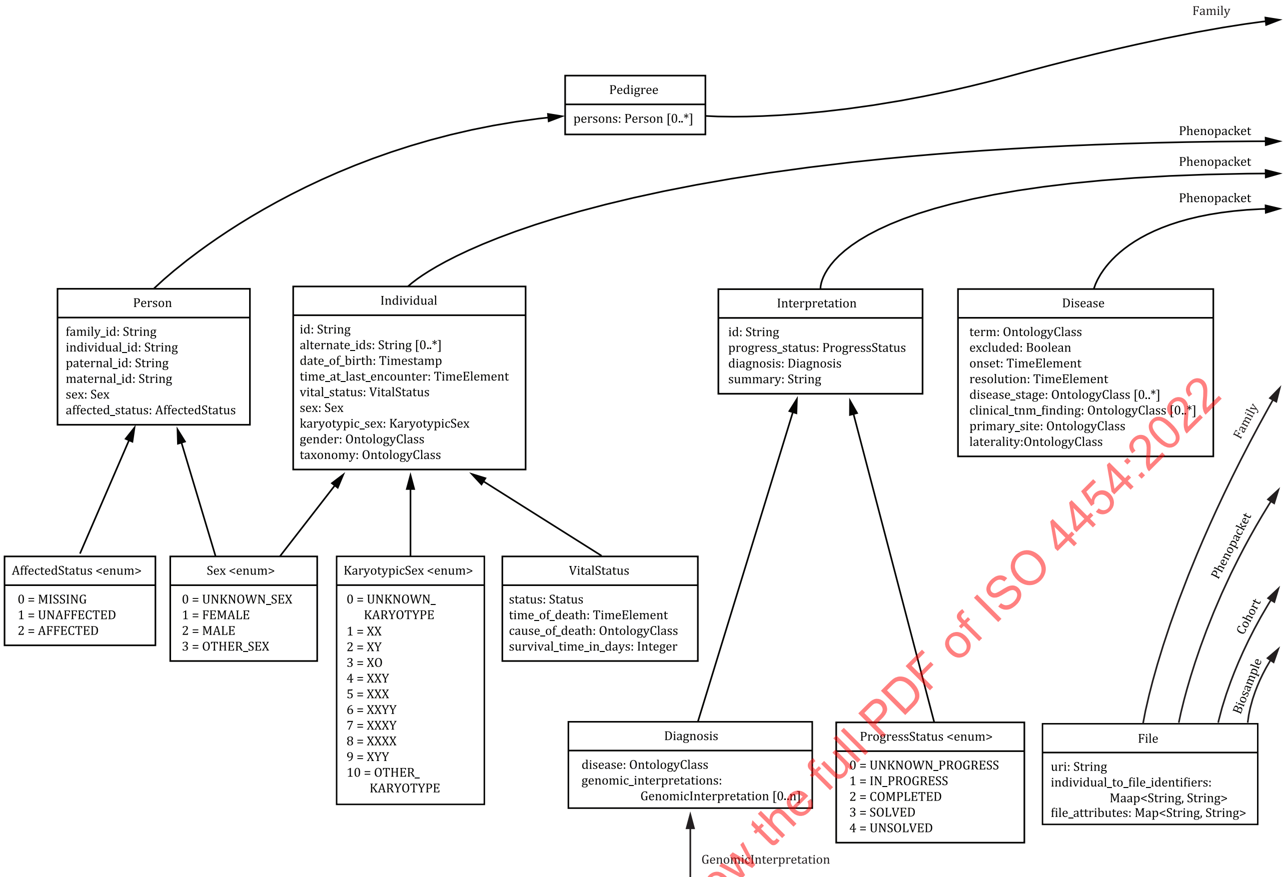
Arrows between elements indicate composition. Some classes have been omitted for legibility. See Figure 2 for a detailed class diagram of each thematic grouping.

Figure 1 — Simplified Overview of Phenopackets Schema Elements


Family
id: String [1..1]
proband: Phenopacket [1..1]
relatives: Phenopacket [0..*]
consanguinous_parents: Boolean
pedigree: Pedigree [1..1]
files: File [0..*]
meta_data: MetaData [1..1]
Cohort
id: String [1..1]
description: String [0..1]
members: Phenopacket [0..*]
files: File [0..*]
meta_data: MetaData [1..1]
Pedigree
Phenopacket
id: String [1..1]
subject: Individual [1..1]
phenotypic_features: PhenotypicFeature [0..*]
measurements: Measurement [0..*]
biosamples: Biosample [0..*]
interpretations: interpretation [0..*]
diseases: Disease [0..*]
medical_actions: MedicalAction [0..*]
files: File [0..*]
meta_data: MetaData [1..1]
Individual
Interpretation
Disease
Medical Action
File
File
File
File
Biosample
id: String
individual_id: String
derived_from_id: String
description: String
sampled_tissue: OntologyClass
sample_type: OntologyClass
phenotypic_features: PhenotypicFeature [0..*]
measurements: Measurement [0..*]
taxonomy: OntologyClass
time_of_collection: TimeElement
histological_diagnosis: OntologyClass
tumor_progression: OntologyClass
tumor_grade: OntologyClass
pathological_stage: OntologyClass
pathological_tnm_finding: OntologyClass [0..*]
diagnostic_markers: OntologyClass [0..*]
procedure: Procedure
files: File [0..*]
material_sample: OntologyClass
sample_processing: OntologyClass
sample_storage: OntologyClass
MetaData
created: google.protobuf.Timestamp
created_by: String
submitted_by: String
resources: Resource [0..*]
updates: Update [0..*]
phenopacket_schema_version: String
external_references: ExternalReference [0..*]
External-
Reference
Procedure
Measurement
description: String
assay: OntologyClass
measurement_value: { Value | ComplexValue}
time_observed: TimeElement
procedure: Procedure
PhenotypicFeature
description: String
type: OntologyClass
excluded: Boolean
severity: OntologyClass
modifiers: OntologyClass [0..*]
onset: TimeElement
resolution: TimeElement
evidence Evidence [0..*]
Update
timestamp: Timestamp
updated_by: String
comment: String
Resource
id: String
name: String
url: String
version: String
namespace_prefix: String
iri_prefix: string
Procedure
Value
value: {Quantity | OntologyClass}
ComplexValue
typed_quantity: TypedQuantity
Evidence
evidence_code: OntologyClass
reference: ExternalReference
ExternalReference
Quantity
TypedQuantity

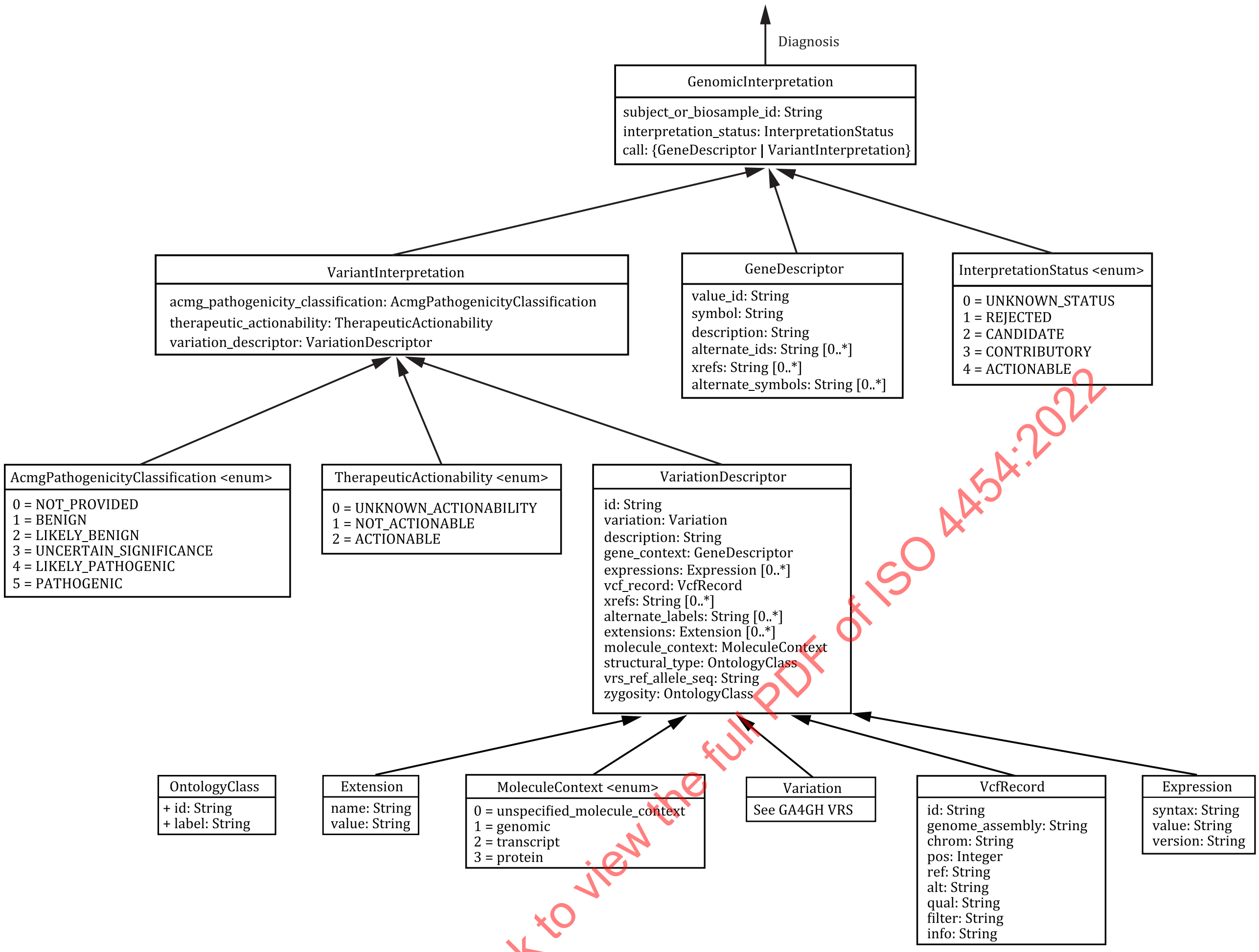
Arrows between elements indicate composition. The OntologyClass and TimeElement links have been omitted for legibility.

a) Detailed view of the top-level base classes of phenopackets, biosample classes and measurement classes



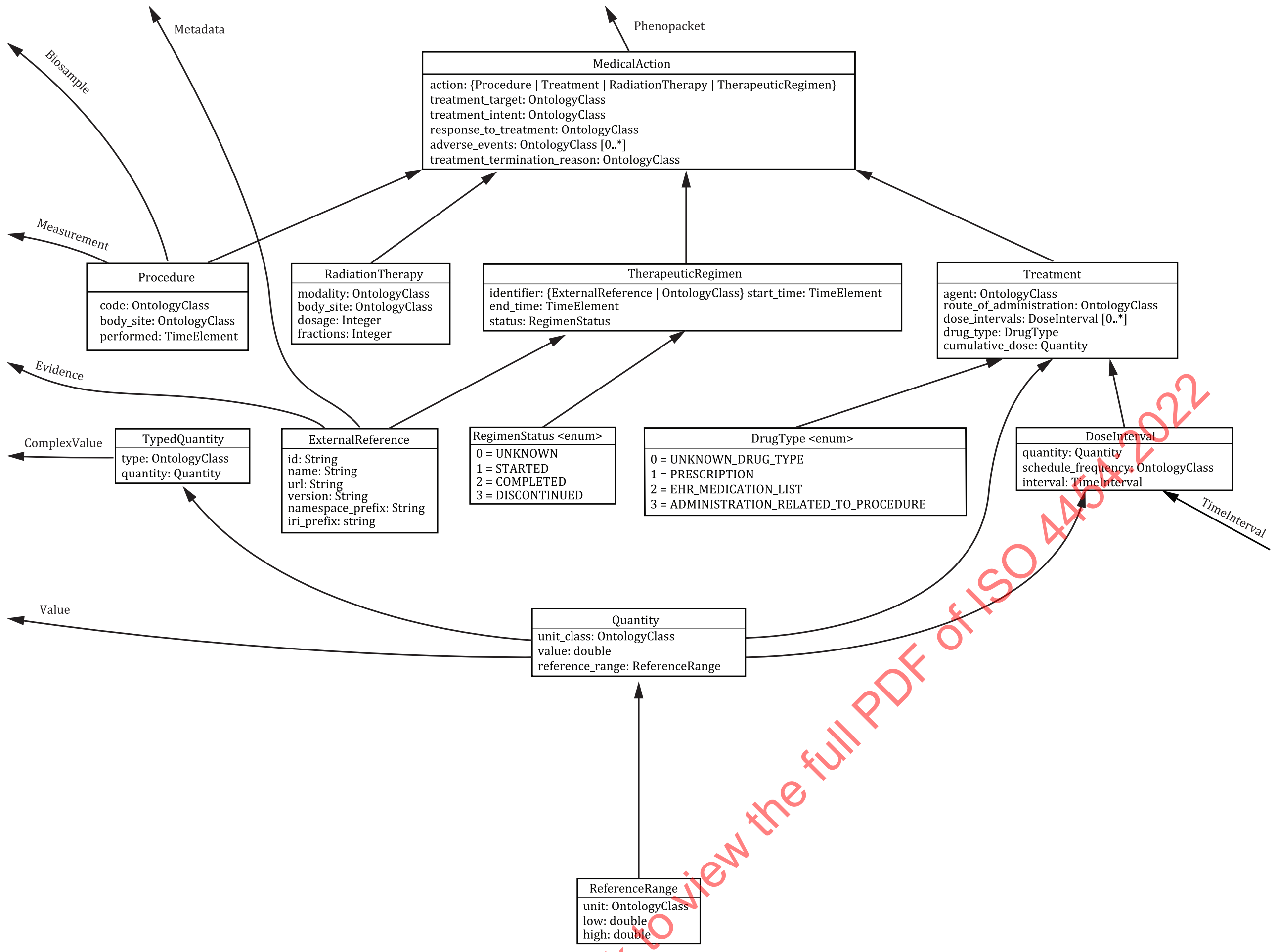
Arrows between elements indicate composition. The OntologyClass and TimeElement links have been omitted for legibility.

b) Detailed view of the base classes and disease/interpretation classes of the phenopackets schema



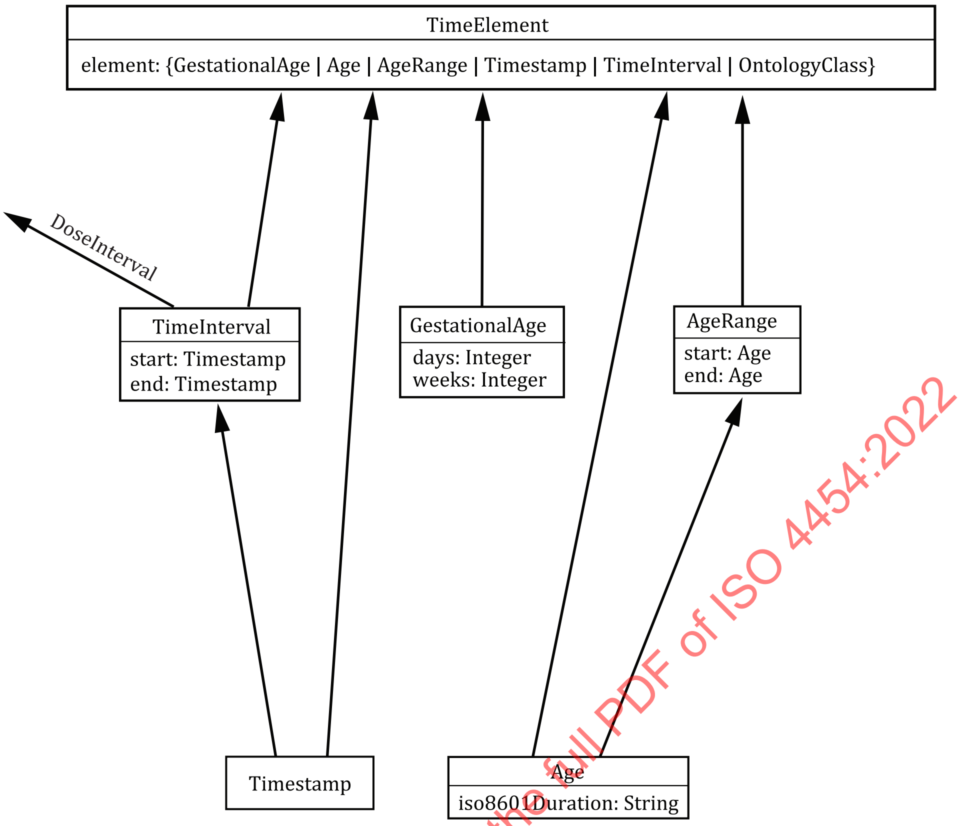
Arrows between elements indicate composition. The OntologyClass links have been omitted for legibility.

c) Detailed view of the genomic interpretation classes of the phenopackets schema



Arrows between elements indicate composition. The OntologyClass and some TimeElement links have been omitted for legibility.

d) Detailed view of the medical action classes of the phenopackets schema



Arrows between elements indicate composition. The OntologyClass links have been omitted for legibility.

e) Detailed view of the multi-purpose time classes of the phenopackets schema

Figure 2 — Phenopackets Schema Class Diagram

5.2 Requirement Levels

5.2.1 General

In protobuf3, as all elements are optional, there is no mechanism within protobuf to declare that a certain field is required. The phenopacket schema does require some fields to be present and, in some cases, additionally requires that these fields have a certain format (syntax) or intended meaning (semantics). Software that uses phenopackets should check the validity of the data with other means.

The requirement levels that are shown for the various elements of the phenopacket only apply if the element is used. For instance, the quantity building block shows that the unit and value fields are required (the multiplicity is exactly 1). In contrast, the field reference_range is optional (the multiplicity can be 0 or 1). The requirements only apply if a quantity element is used in a phenopacket. For instance, phenopackets that do not contain measurement or treatment elements do not contain quantity elements, and so the requirements for the fields in a quantity element do not apply.

5.2.2 Multiplicity

Each building block model contains a multiplicity column. The multiplicity value definitions are as follows:

0..1: The element can be absent (0) or present (1). The element is optional.

1..1: The element shall be present (1). The element shall be used.

0..*: There can be from zero to an arbitrary number of elements, i.e. a potentially empty list. This element is optional.

1..*: There can be from one to an arbitrary number of elements, i.e. a list that shall not be empty. This element shall be used.

5.3 Ontology Use

The phenopacket schema shall use a common ontology that allows sophisticated algorithmic analysis over medically relevant abnormalities. Multiple ontologies can be used within a phenopacket. See Annex B for example ontologies and terms (see 3.21).

EXAMPLE HPO, NCIT, SNOMED

6 Phenopacket Schema Top-Level Elements

6.1 Phenopacket

6.1.1 General

The Phenopacket top-level element can be used to describe the phenotypic characteristics observed in an individual with a disease that is being studied or for an individual in whom the diagnosis is being sought (see Table 1). A phenopacket can contain information about genetic findings that are causative of the disease, or alternatively it can contain a reference to a VCF file if whole exome sequencing is being performed as a part of the differential diagnostic process. A phenopacket can also be used to describe the constitutional phenotypic findings of an individual with cancer. See Annex C, D, and E for example phenopackets generated for rare disease, cancer, and COVID-19 use cases, respectively.

Table 1 — Phenopacket elements

Field	Type	Multiplicity	Description
id	string	1..1	Arbitrary identifier
subject	Individual	0..1	The proband
phenotypic_features	PhenotypicFeature (list)	0..*	Phenotypic features observed in the proband
measurements	Measurement (list)	0..*	Measurements performed on the proband
biosamples	Biosample	0..*	Samples (e.g. biopsies), if any
interpretations	Interpretation (list)	0..*	Interpretations related to the phenopacket
diseases	Disease (list)	0..*	Disease(s) diagnosed in the proband
medical_actions	MedicalAction (list)	0..*	Medical actions performed on the proband
files	File (list)	0..*	A list of files related to the subject
meta_data	MetaData	1..1	Information about ontologies and references used in the phenopacket

6.1.2 id

The id is an identifier specific for this phenopacket. The syntax of the identifier is application specific.

6.1.3 subject

This is the individual human (or another organism) that the phenopacket is describing. In many cases, the individual will be a patient or proband of the study. See 7.15 (Individual) for further information.

6.1.4 phenotypic_features

This is a list of phenotypic findings observed in the subject. See 7.23 (PhenotypicFeature) for further information.

6.1.5 measurements

A list of measurements performed in the patient. In contrast to 7.23 (PhenotypicFeature), which relies on 7.21 (OntologyClass) to specify the observation, the measurement element can be used to report quantitative as well as ordinal or categorical measurements. See 7.18 (Measurement) for further information.

6.1.6 biosamples

This field describes samples that have been derived from the patient who is the object of the phenopacket, or a collection of biosamples in isolation. See 7.4 (Biosample) for further information.

6.1.7 interpretation

An optional list of interpretations related to the phenopacket. This element is intended to represent interpretation of disease or phenotypic findings based on genomic findings and shall relate to either a genetic or genomic investigation of organismal origin (e.g. germline DNA derived from a blood sample) or from a biosample. See 7.16 (Interpretation) for further information.

6.1.8 diseases

This is a field for disease identifiers and can be used for listing either diagnosed or suspected conditions. The resources using these fields should define what this represents in their context. See 7.6 (Disease) for further information.

6.1.9 medical_actions

A list of treatments or other medical actions performed for proband or individual represented by the phenopacket. See 7.19 (MedicalAction) for further information.

6.1.10 files

This element contains a list of pointers to the relevant file(s) for the subject. For example, this could include VCF or the results of another high-throughput sequencing experiment. See 7.11 (File) for further information.

6.1.11 meta_data

This element contains structured definitions of the resources and ontologies used within the Phenopacket. It is expected that every valid Phenopacket contains a MetaData element. See 7.20 (MetaData) for further information.

6.2 Family

6.2.1 General

The optional Family element is used to represent phenotype, sample and pedigree data required for a genomic diagnosis (see Table 2). In many cases, genetic diagnostics of Mendelian and other disease is performed on a family basis in order to check for co-segregation of candidate variants with a disease. Usually, one family member is designated as the proband, for instance, a child affected by a genetic disease can be the proband in a family. Genomic diagnostics can involve genome sequencing of the child and his or her parents. In this case, the Family element would include a Phenopacket for the child (proband field). If the parents themselves display phenotypic findings relevant to the analysis, then Phenopacket elements are included for them (using the relatives field). If the parents do not display any relevant phenotypic findings, then it is not necessary to include Phenopacket elements for them. Instead, their status as unaffected can be recorded with the Pedigree element.

Table 2 — Family elements

Field	Type	Multiplicity	Description
id	string	1..1	Application-specific identifier
proband	Phenopacket	1..1	The proband (index patient) in the family
relatives	Phenopacket (list)	0..*	List of phenopackets for family members other than the proband
pedigree	Pedigree	1..1	Representation of the pedigree
files	File (list)	0..*	A list of files related to the whole family, e.g. multi-sample high-throughput sequencing files
meta_data	MetaData	1..1	Metadata about ontologies and references used in this message

6.2.2 id

An identifier specific for this family.

6.2.3 proband

The individual representing the focus of this packet. See 6.1 (Phenopacket) for further information.

6.2.4 relatives

Individuals related in some way to the patient. The individuals can be genetically related or can be members of a cohort. If this field is used, then it is expected that a pedigree will be included for genetically related individuals for use cases such as genomic diagnostics. If a phenopacket is being used to describe one member of a cohort, then in general one Phenopacket will be created for each of the individuals in the cohort. If all that is required is to record affected/not affected status in a family, it is possible to use the Pedigree element only.

6.2.5 pedigree

The pedigree defining the relations between the proband and their relatives. This element contains information compatible with the information in a PED file. The individual_id in the Pedigree element shall map to the id in the Phenopacket element. See 7.22 (Pedigree) for further information.

6.2.6 files

This element contains a list of pointers to the relevant file(s) for the family as a whole. The file(s) shall refer to the entire family. Otherwise, individual files shall be contained within their appropriate scope, for example, within a Phenopacket for germline samples of an individual or within the scope of the Biosample element in the case of data derived from that biosample.

In the case of multi-sample high-throughput sequencing files the sample identifiers in the high-throughput sequencing file shall each map to individual_id referenced in the Pedigree element, in order that linkage analysis can be performed on the sample. See 7.11 (File) for further information.

6.2.7 meta_data

This element contains structured definitions of the resources and ontologies used within the phenopacket. It is expected that every valid Phenopacket contains a MetaData element. See 7.20 (MetaData) for further information.

6.3 Cohort

6.3.1 General

This element describes a group of individuals related in some phenotypic or genotypic aspect. For instance, a cohort may consist of individuals all of whom have been diagnosed with a certain disease or have been found to have a certain phenotypic feature. See Table 3 for the fields in the Cohort top-level element.

It is recommended to use the Family element to describe families being investigated for the presence of a Mendelian disease.

Table 3 — Cohort elements

Field	Type	Multiplicity	Description
id	string	1..1	Arbitrary identifier for the cohort
description	string	0..1	Arbitrary text description of the cohort
members	Phenopacket (list)	1..*	Phenopackets that represent members of the cohort
files	File (list)	0..*	A list of files related to the whole cohort (e.g. multi-sample high-throughput sequencing files)
meta_data	MetaData	1..1	Metadata related to the ontologies and references used in this message

6.3.2 id

The id is an identifier specific for this cohort. The syntax of the identifier is application specific.

6.3.3 description

Any information relevant to the study or cohort may be added here as free text.

6.3.4 members

One Phenopacket is included for each member of the cohort. See 6.1 (Phenopacket) for more information.

6.3.5 files

This element contains a list of pointers to the relevant HTS file(s) for the cohort. The file(s) shall be a multi-sample file referring to the entire cohort, if appropriate. Otherwise, individual files shall be contained within their appropriate scope, for example, within a Phenopacket for germline samples of an individual or within the scope of a Biosample in the case of data derived from sequencing that biosample. See 7.11 (File) for further information.

6.3.6 meta_data

This element contains structured definitions of the resources and ontologies used within the phenopacket. It is expected that every valid Phenopacket contains a MetaData element. See 7.20 (MetaData) for further information.

7 Phenopackets Building Blocks

7.1 General

The phenopacket standard consists of several protobuf messages each of which contains information about a certain topic such as phenotype, variant, and pedigree. One message can contain other messages, which allows a rich representation of data. For instance, the phenopacket message contains messages of type Individual, PhenotypicFeature, Biosample, and so on. Individual messages can therefore be regarded as building blocks that are combined to create larger structures. It would also be straightforward to include the phenopackets schema into larger schema for particular use cases.

7.2 Age

The Age element allows the age of the subject to be encoded in several different ways that support different use cases. Age is encoded as ISO 8601 duration, in accordance with the ISO 8601 series. See Table 4 for the fields in an Age element.

Table 4 — Age data model

Field	Type	Multiplicity	Description
iso8601duration	string	1..1	An ISO 8601 string representing age

If the Age message is used, the iso8601duration value shall be present. The string element (string age=1) should be used for ISO 8601 durations (e.g. P40Y10M05D). For many use cases, less precise strings will be preferred for privacy reasons, e.g. P40Y.

EXAMPLE

```
age:
  iso8601duration: "P25Y3M2D"
```

7.3 AgeRange

The AgeRange element is intended to be used when the age of a subject is represented by a bin, e.g. 5 years old to 10 years old. Bins such as this are used in some situations in order to protect the privacy of study participants, whose age is then represented by bins such as 45 years old to 49 years old, 50 years old to 54 years old, 55 years old to 59 years old, and so on. See Table 5 for the fields in an AgeRange element.

Table 5 — AgeRange data model

Field	Type	Multiplicity	Description
start	Age	1..1	An Age message
end	Age	1..1	An Age message

EXAMPLE To represent the bin 45 years old to 49 years old, one can use an Age element with **P45Y** as the start element of the AgeRange element, and an Age element with **P49Y** as the end element. An AgeRange.end shall occur after AgeRange.start.

```
ageRange:
  start:
      iso8601duration: "P45Y"
  end:
      iso8601duration: "P49Y"
```

7.4 Biosample

7.4.1 General

This element represents a biosample (see Table 6). Examples would be a tissue biopsy, a single cell from a culture for single cell genome sequencing or a protein fraction from a gradient centrifugation. Several instances (e.g. technical replicates) or types of experiments (e.g. genomic array as well as RNA-seq experiments) can refer to the same biosample.

Table 6 — Biosample data model

Field	Type	Multiplicity	Description
id	string	1..1	Arbitrary identifier
individual_id	string	0..1	Arbitrary identifier
derived_from_id	string	0..1	Id of the biosample from which the current biosample was derived (if applicable)
description	string	0..1	Arbitrary text
sampled_tissue	OntologyClass	0..1	Tissue from which the sample was taken
sample_type	OntologyClass	0..1	The type of biosample (e.g. RNA, DNA, Cultured cells)
phenotypic_features	PhenotypicFeature (list)	0..*	List of phenotypic abnormalities of the sample
measurements	Measurement (list)	0..*	List of measurements related to the sample
taxonomy	OntologyClass	0..1	Species of the sampled individual
time_of_collection	Time Element	0..1	Age of the proband at the time the sample was taken
histological_diagnosis	OntologyClass	0..1	Disease diagnosis that was inferred from the histological examination
tumor_progression	OntologyClass	0..1	Indicates primary, metastatic, recurrent
tumor_grade	OntologyClass (list)	0..1	List of terms to indicate the grade of the tumor
pathological_stage	OntologyClass	0..1	Pathological stage of the tumor, if applicable
Pathological_tnm_finding	OntologyClass (list)	0..*	Pathological TNM findings, if applicable
diagnostic_markers	OntologyClass	0..*	Clinically relevant biomarkers
procedure	Procedure	0..1	The procedure used to extract the biosample
hts_files	HtsFile (list)	0..*	List of high-throughput sequencing files derived from the biosample
material_sample	OntologyClass	0..1	Status of the specimen (normal control, tumor tissue, etc.)
sample_processing	OntologyClass	0..1	How the specimen was processed
sample_storage	OntologyClass	0..1	How the specimen was stored

EXAMPLE The staging system most often used for bladder cancer is the AJCC TNM system. The overall stage is assigned based on the T, N, and M categories (Cancer stage grouping). For instance, stage II (pathological staging) is defined as T2a or T2b, N0, and M0, meaning the cancer has spread into the wall of the bladder.

```
biosample:
  id: "sample1"
  individualId: "patient1"
  description: "Additional information can go here"
  sampledTissue:
      id: "UBERON_0001256"
      label: "wall of urinary bladder"
  histologicalDiagnosis:
      id: "NCIT:C39853"
      label: "Infiltrating Urothelial Carcinoma"
  tumorProgression:
      id: "NCIT:C84509"
      label: "Primary Malignant Neoplasm"
  tumorGrade:
      id: "NCIT:C36136"
      label: "Grade 2 Lesion"
  procedure:
      code:
          id: "NCIT:C5189"
          label: "Radical Cystoprostatectomy"
  files:
      - uri: "file:///data/genomes/urothelial_ca_wgs.vcf.gz"
      individualToFileIdentifiers:
          patient1: "NA12345"
      fileAttributes:
          description: "Urothelial carcinoma sample"
          htsFormat: "VCF"
          genomeAssembly: "GRCh38"
  materialSample:
      id: "EFO:0009655"
      label: "abnormal sample"
  timeOfCollection:
      age:
          iso8601duration: "P52Y2M"
  pathologicalStage:
      id: "NCIT:C28054"
      label: "Stage II"
  pathologicalTnmFinding:
  - id: "NCIT:C48726"
      label: "T2b Stage Finding"
  - id: "NCIT:C48705"
      label: "N0 Stage Finding"
  - id: "NCIT:C48699"
      label: "M0 Stage Finding"
```

7.4.2 id

The biosample ID. This is unique in the context of the server instance.

7.4.3 individual_id

The ID of the Individual this biosample was derived from. It is recommended to provide this information here if the Biosample element is being transmitted as a part of a phenopacket.

7.4.4 derived_from_id

The ID of the parent biosample this biosample was derived from, if applicable.

7.4.5 description

The biosample's description. This attribute contains human readable text. The "description" attributes should not contain any structured data.

7.4.6 sampled_tissue

An OntologyClass describing the tissue from which the specimen was collected. UBERON should be used. The PDX-MI mapping is Specimen tumor tissue.

7.4.7 sample_type

An OntologyClass describing the type of same sample (e.g. RNA, DNA, or Cultured cells). It is recommended to use an EFO term to describe the sample, for instance, genomic DNA (EFO:0008479).

7.4.8 phenotypic_features

The phenotypic characteristics of the biosample, for example histological findings of a biopsy. See 7.23 (PhenotypicFeature) for further information.

7.4.9 measurements

Measurements (usually quantitative) performed on the sample. See 7.18 (Measurement) for further information.

7.4.10 taxonomy

An OntologyClass representing the taxonomic identifier of the organism. For resources where there is more than one organism being studied it is advisable to indicate the taxonomic identifier to its most specific level. It is recommended to use codes from the NCBI Taxonomy resource.

EXAMPLE NCBITaxon:9606 (homo sapiens), NCBITaxon:9615 (dog).

7.4.11 time_at_collection

An age object describing the age of the individual this biosample was derived from at the time of collection. The Age object allows the encoding of the age either as ISO 8601 duration or time interval (preferred), or as ontology term object. See 7.31 (TimeElement) for further information.

7.4.12 histological_diagnosis

This is the pathologist's diagnosis and can represent a refinement of the clinical diagnosis (which could be reported in the Phenopacket that contains this biosample). Normal samples would be tagged with the term "NCIT:C38757", "Negative Finding". See 7.21 (OntologyClass) for further information.

7.4.13 tumor_progression

This field can be used to indicate if a specimen is from the primary tumor, a metastasis or a recurrence. There are multiple ways of representing this using ontology terms, and the terms chosen should have a specific meaning that is application specific.

EXAMPLE A term from the following NCIT terms from the Neoplasm by Special Category can be chosen: Primary Neoplasm, Metastatic Neoplasm or Recurrent Neoplasm

7.4.14 tumor_grade

This should be a child term of NCIT:C28076 (Disease Grade Qualifier) or equivalent.

7.4.15 pathological_stage

An OntologyClass to represent the pathological stage of the sample, if applicable.

7.4.16 pathological_tnm_finding

A list of OntologyClass to represent the pathological TNM findings, if applicable. The TNM staging system is used to describe the amount and spread of cancer in a patient's body.

7.4.17 diagnostic_markers

A field for clinically relevant bio markers. Most of the assays, such as immunohistochemistry, are covered by the NCIT under the sub-hierarchy NCIT:C25294 (Laboratory Procedure).

EXAMPLE NCIT:C68748 (HER2/Neu Positive), NCIT:C131711 (Human Papillomavirus-18 Positive).

7.4.18 procedure

The clinical procedure performed on the subject in order to extract the biosample. See Procedure (7.24) for further information.

7.4.19 files

This element contains a list of pointers to relevant file(s) for the biosample. For example, this could include the results of a high-throughput sequencing experiment. See 7.11 (File) for further information.

7.4.20 material_sample

An OntologyClass that can be used to specify the status of the sample. For instance, a status can be used as a normal control, often in combination with another sample that is thought to contain a pathological finding.

EXAMPLE EFO:0009654 (reference sample), EFO:0009655 (abnormal sample).

7.4.21 sample_processing

An OntologyClass representing the technique used to process the sample.

7.4.22 sample_storage

An OntologyClass representing how the sample was stored.

7.5 ComplexValue

7.5.1 General

This element is intended to represent complex measurements, such as blood pressure, where more than one component quantity is required to describe the measurement (see Table 7).

Table 7 — ComplexValue data model

Field	Type	Multiplicity	Description
Typed_quantities	TypedQuantity	1..*	A list of quantities required to fully describe the complex value.

EXAMPLE The following example shows a ComplexQuantity message for diastolic blood pressure. The intended use case for a ComplexQuantity message is as a component of a Measurement message that contains two or more components (e.g. systolic and diastolic blood pressure).

```
complexValue:
    typedQuantities:
    - type:
        id: "NCIT:C25298"
        label: "Systolic Blood Pressure"
      quantity:
        unit:
            id: "NCIT:C49670"
            label: "Millimeter of Mercury"
      value: 120.0
    - type:
        id: "NCIT:C25299"
        label: "Diastolic Blood Pressure"
      quantity:
        unit:
            id: "NCIT:C49670"
            label: "Millimeter of Mercury"
        value: 70.0
```

7.5.2 typed_quantities

The ComplexValue element consists of a list of TypedQuantity elements, describing the type of measurement and the outcome (see Table 8).

Table 8 — TypedQuantity data model

Field	Type	Multiplicity	Description
type	OntologyClass	1..1	An ontology class to describe the type of measurement
quantity	Quantity	1..1	A field to denote the outcome of the measurement

7.5.2.1 type

An ontology class to describe the type of measurement, including an identifier and associated label

EXAMPLE Systolic blood pressure (NCIT:C25298).

7.5.2.2 quantity

A field to denote the outcome of the measurement. See 7.25 (Quantity) for more information.

7.6 Disease

7.6.1 General

Disease refers to a diagnosis, an inference or hypothesis about the cause underlying the observed phenotypic abnormalities. See Table 9 for the fields in the Disease element.

Table 9 — Disease data model

Field	Type	Multiplicity	Description
term	OntologyClass	1..1	An ontology class that represents the disease
excluded	boolean	1..1	A flag to indicate whether the disease was observed or not
onset	Time Element	0..1	The age of onset of the disease
resolution	Time Element	0..1	The age of resolution (abatement) of the disease
disease_stage	OntologyClass (list)	0..*	List of terms representing the disease stage
clinical_tnm_finding	OntologyClass (list)	0..*	List of terms representing the tumor TNM score
primary_site	OntologyClass	0..1	The primary site of the disease
laterality	OntologyClass	0.1	The laterality (left or right) of the primary site, if applicable

EXAMPLE

```
disease:
  term:
      id: "OMIM:164400"
      label: "Spinocerebellar ataxia 1"
  onset:
      age:
          iso8601duration: "P38Y7M"
```

7.6.2 term

The disease element denotes the diagnosis by means of a term, or ontology class. It is recommended to use a term from OMIM, Orphanet, MONDO, Disease Ontology, SNOMED, ICD, or NCIT.

EXAMPLE OMIM:101600, Orphanet:710, MONDO:0007043, DOID:0080009, NCIT:C9049.

7.6.3 excluded

A flag to indicate whether the disease was observed or not. The default is ‘false’, or in other words, the disease was observed. Therefore, it is only required in cases to indicate that the disease was looked for but found to be absent. More formally, this modifier indicates the logical negation of the OntologyClass used in the ‘term’ field.

7.6.4 onset

The onset element uses a TimeElement to describe the onset of the disease. See 7.31 (TimeElement) for more information. It is also possible to denote the onset of individual phenotypic features of disease in the Phenopacket element. If an ontology class is used to refer to the age of onset of the disease, it is recommended to use a term from the HPO onset hierarchy.

7.6.5 resolution

An element representing the age of resolution (e.g. abatement, recovery from) of the disease. See 7.31 (TimeElement) for more information.

7.6.6 disease_stage

This attribute is used to describe the stage of disease. If the disease is a cancer, this attribute describes the extent of cancer development, typically including an AJCC stage group (i.e. Stage 0, I to IV), though other staging systems are used for some cancers. The list of elements constituting this attribute should be derived from child terms of NCIT:C28108 (Disease Stage Qualifier) or equivalent hierarchy from another ontology.

7.6.7 clinical_tnm_finding

This attribute can be used if the phenopacket is describing cancer. TNM findings score the progression of cancer with respect to the originating tumor (T), spread to lymph nodes (N), and presence of metastases (M). These findings are commonly reported for tumors and support the stage classifications stored in the disease_stage attribute. The list of elements constituting this attribute should be derived from child terms of NCIT:C48232 (Cancer TNM Finding) or equivalent hierarchy from another ontology.

7.6.8 primary_site

The term used to describe the primary site of the disease. It is recommended to use terms from NCIT or UBERON.

7.6.9 laterality

A term used to indicate laterality (left or right) of the primary site of diagnosis, if applicable.

7.7 DoseInterval

7.7.1 General

This element represents a block of time in which the dosage of a medication was constant. For example, to represent a period of 30 mg twice a day for an interval of 10 days, a Quantity element can be used to represent the individual 30 mg dose, and OntologyClass element to represent twice a day, and an interval element to represent the 10-day interval. See Table 10 for the fields in a DoseInterval element.

Table 10 — DoseInterval data model

Field	Type	Multiplicity	Description
quantity	Quantity	1..1	The amount administered in one dose
schedule_frequency	OntologyClass	1..1	How often doses are administered per indicated duration (e.g. per day)
interval	TimeInterval	1..1	The specific interval over which the dosage was administered in the indicated quantity

EXAMPLE The following message represents a dose interval from March 15, 2020 to March 25, 2020, in which a constant dose of 30 mg was given twice a day.

```
doseInterval:
  quantity:
      unit:
          id: "UO:0000022"
          label: "milligram"
      value: 30.0
  scheduleFrequency:
      id: "NCIT:C64496"
      label: "Twice Daily"
  interval:
      start: "2020-03-15T13:00:00Z"
      end: "2020-03-25T09:00:00Z"
```

7.7.2 quantity

The amount of an individual dose. See 7.25 (Quantity) for more information.

7.7.3 schedule_frequency

This element specifies the number of instances within a specific time period. It is intended to have the same meaning as the NCIT schedule frequency class.

7.7.4 interval

The time interval over which the specified dosage is given. See 7.31 (TimeInterval) for information about privacy concerns.

7.8 DrugType

7.8.1 General

DrugType is an enumerated element to describe the context of drug administration, within a Treatment message (see 7.34). This element does not intend to capture information about the administration route (e.g. by mouth or intravenous) or the medical reason for administering a treatment but rather about the setting - inpatient, outpatient, or related to a (generally one-time) procedure. It is assumed that medications recorded on the medication list of an EHR are likely to have been administered as prescribed. If a prescription is given for outpatient use, it is less likely that the medication will be taken exactly as prescribed[8]. Finally, medications can be given to conduct a medical procedure such as a local anaesthetic given before a skin biopsy or a sedative given to perform a bronchoscopy. See Table 11 for the DrugType enumeration values.

Table 11 — DrugType enumeration values

Item	Value
UNKNOWN_DRUG_TYPE	0
PRESCRIPTION	1
EHR_MEDICATION_LIST	2
ADMINISTRATION_RELATED_TO_PROCEDURE	3

7.9 Evidence

7.9.1 General

This element intends to represent the evidence for an assertion such as an observation of a PhenotypicFeature (see Table 12). Terms from the ECO should be used.

Table 12 — Evidence data model

Field	Type	Multiplicity	Description
evidence_code	OntologyClass	1..1	An ontology class that represents the evidence type
reference	ExternalReference	0..1	Representation of the source of the evidence

EXAMPLE

```
evidence:
    evidenceCode:
        id: "ECO:0006017"
        label: "author statement from published clinical study used in manual assertion"
    reference:
        id: "PMID:30962759"
        description: "Recurrent Erythema Nodosum in a Child with a SHOC2 Gene Mutation"
```

7.9.2 evidence_code

An ontology class that represents the evidence type.

EXAMPLE In order to describe the evidence for a phenotypic observation that is derived from a publication, one can use the ECO term author statement from published clinical study used in manual assertion (ECO: 0006017) and record a PubMed id in the reference field.

7.9.3 Reference

A representation of the source of the evidence. An ExternalReference is used to store a reference to the publication or other source that supports the evidence. Not all types of evidence will have an external reference, and therefore this field is optional.

7.10 ExternalReference

7.10.1 General

This element encodes information about an external reference (see Table 13). One typical use case for this element is to provide a reference to a published article by including a PubMed identifier in the Evidence element.

Table 13 — ExternalReference data model

Field	Type	Multiplicity	Description
id	string	0..1	An application specific identifier
reference	string	0..1	An application specific identifier
description	string	0..1	An application specific description

EXAMPLE

```
externalReference:
    id: "PMID:30962759"
    description: "Recurrent Erythema Nodosum in a Child with a SHOC2 Gene Mutation"

externalReference:
    id: "PMID:30962759"
    reference: "https://pubmed.ncbi.nlm.nih.gov/30962759"
    description: "Recurrent Erythema Nodosum in a Child with a SHOC2 Gene Mutation"
```

7.10.2 id

The syntax of the identifier is application specific. This should be a CURIE that uniquely identifies the evidence source, URL/URI, or other relevant identifier. A CURIE identifier should be used, and if one is used, the corresponding Resource should be provided in the MetaData element.

EXAMPLE ISBN:978-3-16-148410-0 or PMID:123456.

7.10.3 reference

A full or partial URL/URI should be provided for systems to resolve an external reference, especially in the absence of a CURIE identifier.

7.10.4 description

An optional free text description of the evidence.

7.11 File

7.11.1 General

The File message allows a phenopacket to link the structured phenotypic data it contains to external files which may be used to inform analyses (see Table 14). For example, the file could link to sequencing data in VCF format as well as other data types.

Given that File elements are listed in various locations such as the Phenopacket, Biosample, Family, etc. which can in turn be nested, individual files shall be contained within their appropriate scope. For example, within a Phenopacket for germline samples of an individual or within the scope of the Biosample in the case of genomic data derived from sequencing or other characterization of that biosample. Aggregate data types such as Cohort and Family shall contain aggregate file data, like a merged/multi-sample VCF at the level of the Family/Cohort, but each member Phenopacket can contain its own locally scoped files pertaining to that individual/biosample(s).

Table 14 — File data model

Field	Type	Multiplicity	Description
uri	string	1..1	A valid URI
individual_to_file_identifiers	A map of string key: value	0..1	The mapping between the Individual.id or Biosample.id to any identifier in the file
file_attributes	A map of string key: value	0..1	A map of attributes pertaining to the file or its contents

EXAMPLE The message below shows how a compressed VCF file can be specified. It indicates the sample identifier NA12345 in the VCF file is equivalent to the Phenopacket individual identifier patient23456. The fileAttributes indicate that the file was called against the GRCh38 genome assembly, indicates the fileFormat is VCF and provides a human-readable description.

```
file:
  uri: "file://data/genomes/germline_wgs.vcf.gz"
  individualToFileIdentifiers:
    patient23456: "NA12345"
  fileAttributes:
    genomeAssembly: "GRCh38"
    fileFormat: "vcf"
    description: "Matched normal germline sample"
```

7.11.2 uri

URI for the file.

EXAMPLE file://data/genomes/file1.vcf.gz or https://opensnp.org/data/60.23andme-exome-vcf.231?1341012444.

7.11.3 individual_to_file_identifiers

A map of identifiers mapping an individual referred to in the phenopacket to an identifier used in the file. The key values shall correspond to the Individual id field for the individuals in the message or Biosample id field for biosamples, the values shall map to identifiers in the file.

7.11.4 file_attributes

A map of attributes that a resource might want to share about the contents of a file. For example, a file containing genomic coordinates (e.g. VCF, BED) should contain an entry with the key genomeAssembly and a value indicating the genome assembly the contents of the file was called against. It is recommended to use the Genome Reference Consortium nomenclature (e.g. GRCh37, GRCh38).

7.12 GeneDescriptor

7.12.1 General

This element represents an identifier for a gene, using the Gene Descriptor from the VRSATILE Framework[9] (see Table 15). Gene Descriptors can be used to transmit the information that the gene is thought to play a causative role in the disease phenotypes being described in cases where the exact variant cannot be transmitted, either for privacy reasons or because it is unknown. Gene Descriptors can also be used to contextualize variants described in 7.37 (VariationDescriptor).

Table 15 — GeneDescriptor data model

Field	Type	Multiplicity	Description
value_id	string	1..1	Official identifier of the gene
symbol	string	1..1	Official gene symbol
description	string	0..1	A free-text description of the gene
alternate_ids	string (list)	0..*	Alternative identifier(s) of the gene
alternate_symbols	string (list)	0..*	Alternative symbol(s) of the gene
xrefs	string (list)	0..*	Relative concept IDs (e.g. gene ortholog IDs) can be placed in in this field

EXAMPLE

```
geneDescriptor:
  valueId: "HGNC:3477"
  symbol: "ETF1"
```

Optionally, with alternate identifiers:

```
geneDescriptor:
  valueId: "HGNC:3477"
  symbol: "ETF1"
  alternateIds:
  - "ensembl:ENSG00000120705"
  - "ncbigene:2107"
  - "ucsc:uc003ldc.6"
  - "OMIM:600285"
```

Using the gene descriptor to convey alternate identifiers, symbols and orthologs:

```
geneDescriptor:
  valueId: "HGNC:3477"
  symbol: "ETF1"
  alternateIds:
  - "ensembl:ENSG00000120705"
  - "ncbigene:2107"
  - "ucsc:uc003ldc.6"
  - "OMIM:600285"
  alternateSymbols:
  - "SUP45L1"
  - "ERF1"
  - "ERF"
  - "eRF1"
  - "TB3-1"
  - "RF1"
  xrefs:
  - "VGNC:97422"
  - "MGI:2385071"
  - "RGD:1305712"
  - "ensembl:ENSRNOG00000019450"
  - "ncbigene:307503"
```

7.12.2 value_id

The id represents the accession number of a comparable identifier for the gene. It should be a CURIE identifier with a prefix that is used by the official organism gene nomenclature committee. In the case of humans, this is the HGNC.

EXAMPLE HGNC:347.

7.12.3 symbol

The official gene symbol. This should use official gene symbol as designated by the organism gene nomenclature committee.

EXAMPLE The HUGO Gene Nomenclature Committee, ETF1.

7.12.4 description

A free-text description of the value object. This should be only used to convey information that is otherwise not possible to encode using the schema.

7.12.5 alternate_ids

This field can be used to provide identifiers to alternative resources where this gene is used or catalogued. These identifiers should be represented in CURIE form with a corresponding Resource.

EXAMPLE ncbigene:2107, ensembl:ENSG00000120705.

7.12.6 alternate_symbols

This field can be used to list the alternate symbols used to refer to the gene. These include the previously approved gene symbols and those used in the literature or other databases to refer to the gene.

7.12.7 xrefs

This field can be used to provide identifiers to alternative resources representing related but not equivalent concepts, for example gene ortholog ids.

7.13 GenomicInterpretation

7.13.1 General

This element is used as a component of the Interpretation element and describes the interpretation for an individual variant or gene (see Table 16). Multiple variants or genes may support the interpretation related to one disease. See 7.16 (Interpretation) for examples.

Table 16 — GenomicInterpretation data model

Field	Type	Multiplicity	Description
subject_or_biosample_id	string	1..1	The id of the patient or biosample that is the subject being interpreted
interpretation_status	Enum InterpretationStatus	1..1	The status of the interpretation
call	Oneof {GeneDescriptor | VariantInterpretation}	1..1	A gene or variant representing the subject of the genomic interpretation

EXAMPLE

```
genomicInterpretation:
    subjectOrBiosampleId: "subject 1"
    interpretationStatus: "CONTRIBUTORY"
    variantInterpretation:
      acmgPathogenicityClassification: "PATHOGENIC"
      variationDescriptor:
        expressions:
        - syntax: "hgvs"
          value: "NM_001848.2:c.877G>A"
        allelicState:
          id: "GENO:0000135"
        label: "heterozygous"
```

7.13.2 subject_or_biosample_id

The identifier of the patient or biosample that is the subject being interpreted. Each genomic interpretation is based on a genomic finding in the germline DNA of the individual referenced in the phenopacket or of a biosample derived from the individual. The identifier used here shall therefore match with the Individual id field or with the Biosample id field.

7.13.3 interpretation_status

An enumeration that describes the conclusion made about the genomic interpretation. The enumerated values are shown in Table 17.

In an autosomal dominant Mendelian disease, one variant is causative. In this case, one would classify it as causative and the Interpretation object that contains the genomic interpretation would classify it as solved. Similarly, in the case of an autosomal recessive disease, one would classify a homozygous variant as causative. There are several situations in which one should use a contributory classification. In the case of an autosomal recessive disease, two contributory genomic interpretations would be used for compound heterozygous variants. In cancer, a contributory classification may be used for multiple variants.

Table 17 — InterpretationStatus enumeration values

Name	Ordinal	Description
UNKNOWN_STATUS	0	No information is available about the status
REJECTED	1	The variant or gene reported here is interpreted not to be related to the diagnosis
CANDIDATE	2	The variant or gene reported here is interpreted to possibly be related to the diagnosis
CONTRIBUTORY	3	The variant or gene reported here is interpreted to be related to the diagnosis
CAUSATIVE	4	The variant or gene reported here is interpreted to be causative of the diagnosis

7.13.4 call

Either a GeneDescriptor or a VariantInterpretation representing the subject of the genomic interpretation (see 7.12 and 7.38, respectively).

7.14 GestationalAge

7.14.1 General

Gestational age is conventionally expressed as completed weeks. Therefore, a 25-week, 5-day fetus is considered a 25-week fetus. Gestational age is often reported as weeks+days. For instance, 33 weeks and 2 days could be reported as 33+2. The gestational age element is intended for use in phenopackets that describe prenatal clinical data. See Table 18 for the fields in the GestationalAge data model.

Table 18 — GestationalAge data model

Field	Type	Multiplicity	Description
weeks	int32	1..1	Completed weeks of gestation (see definition above)
days	int32	0..1	Completed day of gestation, in addition to complete weeks (if available)

EXAMPLE The following shows how the element can be used to report the gestational age of 33 weeks and 2 days.

```
gestationalAge:
    weeks: 33
    days: 2
```

7.15 Individual

7.15.1 General

The subject of the phenopacket is represented by an Individual element. This element intends to represent an individual human or other organism (see Table 19).

Table 19 — Individual data model

Field	Type	Multiplicity	Description
id	string	1...1	An arbitrary identifier
alternate_ids	CURIE (list)	0..*	Alternative identifiers for the individual
date_of_birth	Timestamp	0..1	A timestamp either exact or imprecise
time_at_last_encounter	TimeElement	0..1	The age or age range of the individual when last encountered
vital_status	VitalStatus	0..1	Whether the individual is living or deceased
sex	Sex	0..1	Observed apparent sex of the individual
karyotypic_sex	KaryotypicSex	0..1	The karyotypic sex of the individual
gender	OntologyClass	0..1	The self-identified gender of the individual
taxonomy	OntologyClass	0..1	An OntologyClass representing the species

EXAMPLE The following example is typical but does not make use of all of the optional fields of this element.

```
individual:
  id: "patient:0"
  dateOfBirth: "1998-01-01T00:00:00Z"
  sex: "MALE"
```

7.15.2 id

This element is the primary identifier for the individual and should be used in other parts of a message when referring to this individual. The contents of the element are context dependent and will be determined by the application.

EXAMPLE 1 If the phenopacket is being used to represent a case study about an individual with some genetic disease, the individual can be referred to in that study by their position in the pedigree - III:2 for the second person in the third generation. In this case, id would be set to III:2.

If a Pedigree element is used, the individual_id of the Pedigree element shall match the id field here. If a Biosample element is used, the individual_id of the Biosample element shall match the id field here.

All identifiers within a phenopacket pertaining to an individual should use this identifier. It is the responsibility of the sender to provide the recipient with an internally consistent message. This is possible as all messages can be created dynamically be the sender using identifiers appropriate for the receiving system.

EXAMPLE 2 A hospital might want to send a Family to an external laboratory for analysis. Here, the hospital is providing an obfuscated identifier, which is used to identify the individual in the Phenopacket, the Pedigree and mappings to the sample id in the File. In this case the Pedigree is created by the sending system from whatever source they use, and the identifiers can be mapped to those Individual.id contained in the Family.proband and Family.relatives phenopackets.

In the case of the VCF file, the sending system likely has no control or ability to change the identifiers used for the sample id and it is likely they use different identifiers. For this reason, the File has a local mapping field individual_to_file_identifiers where the Individual id can be mapped to the sample id in that file.

EXAMPLE 3

```
individual:
  id: "patient23456"
  dateOfBirth: "1998-01-01T00:00:00Z"
  sex: "MALE"
```

Assuming that this individual was sequenced, a user can have the following HtsFile element.

```
file:
  uri: "file://data/genomes/germline_wgs.vcf.gz"
  individualToFileIdentifiers:
    patient23456: "NA12345"
  fileAttributes:
    genomeAssembly: "GRCh38"
    fileFormat: "VCF"
    description: "Matched normal germline sample"
```

Example 3 shows individual blocks which would be used as part of a singleton ‘family’ to illustrate the use of the internally consistent Individual.id. As noted above, the data might have been constructed by the sender from different sources but given they know these relationships, they should provide the receiver with a consistent view of the data both for ease of use and to limit incorrect mapping. Thus, one would use the same id various elements. The user would also use patient23456 as the individualId element within a Pedigree element.

7.15.3 alternate_ids

An optional list of alternative identifiers for this individual. These should be in the form of CURIEs and hence have a corresponding resource listed in the MetaData element. These should not be used elsewhere in the phenopacket as this will break the assumptions required for using the id field as the primary identifier. This field is provided for the convenience of users who can have multiple mappings to an individual which they need to track.

7.15.4 date_of_birth

This field represents the date of birth of the individual as an ISO 8601 UTC timestamp that is rounded down to the closest known year/month/day/hour/minute. The element is provided for use cases within protected networks, but it many situations, the element should not be used in order to protect the privacy of the individual. Instead, the Age element should be preferred. See 7.32 (TimeInterval) for more information about timestamps.

EXAMPLE

“2018-03-01T00:00:00Z” for someone born on an unknown day in March 2018.

“2018-01-01T00:00:00Z” for someone born on an unknown day in 2018.

empty if unknown/ not stated.

7.15.5 time_at_last_encounter

An object describing when the encounter with the patient happened or the age of the individual at the time of collection of biospecimens or phenotypic observations reported in the current phenopacket. It is specified using either a TimeElement (see 7.31), which can represent a time in several different ways, either precisely or within a range. For example, this could be an Age or an AgeRange element, which can represent a range of ages such as 10 years old to 14 years old (age may be represented in this way to protect privacy of study participants).

7.15.6 vital_status

The vital status can be used to report whether the individual is living or dead at the timepoint when the phenopacket was created (or if the status is unknown). See 7.39 (VitalStatus) for more information.

7.15.7 sex

Phenopackets make use of an enumeration to denote the phenotypic sex of an individual. See 7.29 (Sex) for more information.

7.15.8 karyotypic_sex

Phenopackets make use of an enumeration to denote the chromosomal sex of an individual. See 7.17 (KaryotypicSex) for more information.

7.15.9 gender

An Ontology Class representing the self-identified gender of the individual.

7.15.10taxonomy

An OntologyClass representing the species. For resources where there can be more than one organism being studied it is advisable to indicate the taxonomic identifier of that organism, to its most specific level. Codes from the NCBI Taxonomy resource should be used.

EXAMPLE NCBITaxon:9606.

7.16 Interpretation

7.16.1 General

This message intends to represent the interpretation of a genomic analysis, such as the report from a diagnostic laboratory (see Table 20).

Table 20 — Interpretation data model

Field	Type	Multiplicity	Description
id	string	1..1	An arbitrary identifier
progress_status	ProgressStatus	1..1	The current resolution status
diagnosis	Diagnosis	0..*	One or more diagnoses, if made
summary	string	0..1	Any additional data about this interpretation

EXAMPLE 1 In this example, a case with id CONSORTIUM:0000123456 is reported to be solved. The diagnosis is Miller syndrome, and the supporting interpretation states the involved gene. For privacy reasons, the variant was not reported, but the intended meaning is that a relevant variant in the named gene was found.

```
interpretation:
  id: "CONSORTIUM:0000123456"
  progressStatus: "SOLVED"
  diagnosis:
    disease:
      id: "OMIM:263750"
      label: "Miller syndrome"
    genomicInterpretations:
    - interpretationStatus: "CONTRIBUTORY"
      geneDescriptor:
        valueId: "HGNC:2867"
        symbol: "DHODH"
```

EXAMPLE 2 Diagnostic finding in an autosomal dominant disease. The Interpretation element might be used to report a laboratory finding in a diagnostic setting or in a published case report. The following example shows how the variant NM_000138.4(FBN1):c.6751T>A (p.Cys2251Ser) would be reported. The subjectOrBiosampleId is set to the id of the individual of the enclosing phenopacket to indicate that the genomic interpretation refers to a germline variant.

```
interpretation:
  id: "Arbitrary interpretation id"
  progressStatus: "SOLVED"
  diagnosis:
    disease:
      id: "OMIM:154700"
      label: "Marfan syndrome"
    genomicInterpretations:
    - subjectOrBiosampleId: "subject 1"
      interpretationStatus: "CONTRIBUTORY"
      variantInterpretation:
        acmgPathogenicityClassification: "PATHOGENIC"
        variationDescriptor:
          expressions:
          - syntax: "hgvs"
            value: "NM_000138.4(FBN1):c.6751T>A"
          allelicState:
            id: "GENO:0000135"
            label: "heterozygous"
```

EXAMPLE 3 Diagnostic finding in a cancer. Cancer cases are not generally solved by genomic analysis. Instead, the intention is often to identify actionable variants that represent potential indications for targeted therapy. In this example, a BRAF variant is interpreted as being actionable. The subjectOrBiosampleId is set to the id of the biosample that is contained in the enclosing phenopacket, representing a biopsy from a melanoma sample taken from the subject of the phenopacket.

```
interpretation:
 id: "Arbitrary interpretation id"
 progressStatus: "COMPLETED"
 diagnosis:
   disease:
     id: "NCIT:C3224"
     label: "Melanoma"
   genomicInterpretations:
   - subjectOrBiosampleId: "biosample id"
     interpretationStatus: "CONTRIBUTORY"
     variantInterpretation:
       acmgPathogenicityClassification: "PATHOGENIC"
       therapeuticActionability: "ACTIONABLE"
       variationDescriptor:
         expressions:
         - syntax: "hgvs"
           value: "NM_001374258.1(BRAF):c.1919T>A (p.Val640Glu)"
         allelicState:
           id: "GENO:0000135"
           label: "heterozygous"
```

7.16.2 id

An arbitrary identifier, however, the id has the same interpretation as the id field in the Individual element.

7.16.3 progress_status

The interpretation has a ProgressStatus that refers to the status of the attempted diagnosis. See Table 21 for the enumerated types.

This is an enumerated type, therefore the values represented below are the only legal values. The value of this type shall not be null, instead it shall use the 0 (zero) ordinal element as the default value, should none be specified.

Table 21 — ProgressStatus enumerated types

Name	Ordinal	Description
UNKNOWN_PROGRESS	0	No information is available about the diagnosis
IN_PROGRESS	1	No diagnosis has been found to date, but additional differential diagnostic work is in progress
COMPLETED	2	The work on the interpretation is complete
SOLVED	3	The interpretation is complete and also considered to be a definitive diagnosis
UNSOLVED	4	The interpretation is complete, but no definitive diagnosis was found

7.16.4 diagnosis

The diagnosis element is meant to refer to the disease that is inferred to be present in the individual or family being analyzed (see Table 22). The diagnosis can be made by means of an analysis of the phenotypic or genomic findings or both. The element is optional because if the resolution_status is UNSOLVED then there is no diagnosis.

Table 22 — Diagnosis data model

Field	Type	Multiplicity	Description
disease	OntologyClass	1..1	The diagnosed condition
genomic_interpretations	GenomicInterpretation	0..*	The genomic elements assessed as being responsible for the disease (may be left empty)

7.16.5 summary

An optional string including additional data about the interpretation.

7.17 KaryotypicSex

This enumeration represents the chromosomal sex of an individual (see Table 23). This is an enumerated type, therefore the values represented below are the only legal values. The value of this type shall not be null, instead it shall use the 0 (zero) ordinal element as the default value, should none be specified.

Table 23 — KaryotypicSex enumeration values

Name	Ordinal	Description
UNKNOWN_KARYOTYPE	0	Untyped or inconclusive karyotyping
XX	1	Female
XY	2	Male
XO	3	Single X chromosome only
XXY	4	Two X and one Y chromosome
XXX	5	Three X chromosomes
XXYY	6	Two X chromosomes and two Y chromosomes
XXXY	7	Three X chromosomes and one Y chromosome
XXXX	8	Four X chromosomes
XYY	9	One X and two Y chromosomes
OTHER_KARYOTYPE	10	None of the above types

7.18 Measurement

7.18.1 General

The Measurement element is used to record individual measurements (see Table 24). It can capture quantitative, ordinal (e.g. absent/present), or categorical measurements.

Table 24 — Measurement data model

Field	Type	Multiplicity	Description
description	string	0..1	A free text description
assay	OntologyClass	1..1	Describes the assay used to produce the measurement
value	one of {Value | ComplexValue}	1..1	The result of the measurement
time_observed	TimeElement	0..1	The time at which the measurement was performed
procedure	Procedure	0..1	The clinical procedure performed to acquire the sample used for the measurement

EXAMPLE 1 The following example shows measurement of platelet count. The result is abnormally low, but in general this element can be used to represent normal or abnormal measurements.

```
measurement:
    assay:
        id: "LOINC:26515-7"
        label: "Platelets [#/volume] in Blood"
    value:
        quantity:
            unit:
                id: "UO:0000316"
                label: "cells per microliter"
            value: 24000.0
    referenceRange:
        unit:
            id: "UO:0000316"
            label: "cells per microliter"
        low: 150000.0
        high: 450000.0
    timeObserved:
        timestamp: "2020-10-01T10:54:20.021Z"
```

EXAMPLE 2 The following example shows an ordinal measurement. The measurement is for nitrite in urine, and the result is positive (present). It would also be appropriate to represent the result of this test as a PhenotypicFeature using the HPO term Nitrituria. Which option to use depends on the analysis goals. The measurement object is intended to represent specific measurements, and the PhenotypicFeature is often used to represent a conclusion that is reached on the basis of the test.

```
measurement:
    assay:
        id: "LOINC:5802-4"
        label: "Nitrite [Presence] in Urine by Test strip"
    value:
        ontologyClass:
            id: "NCIT:C25626"
            label: "Present"
    timeObserved:
        timestamp: "2021-01-01T10:54:20.021Z"
```

EXAMPLE 3 The following example presents a blood pressure measurement. The measurement of blood pressure consists of two measurements (systolic and diastolic), that are represented as a ComplexQuantity.

```
measurement:
    assay:
      id: "CMO:0000003"
      label: "blood pressure measurement"
```

```
complexValue:
  typedQuantities:
  - type:
      id: "NCIT:C25298"
      label: "Systolic Blood Pressure"
    quantity:
        unit:
          id: "NCIT:C49670"
          label: "Millimeter of Mercury"
        value: 125.0
  - type:
      id: "NCIT:C25299"
      label: "Diastolic Blood Pressure"
    quantity:
      unit:
        id: "NCIT:C49670"
        label: "Millimeter of Mercury"
      value: 75.0
timeObserved:
  timestamp: "2021-01-01T10:54:20.021Z"
```

7.18.2 description

A free-text description of the measurement. This is not the appropriate field to document/describe the phenotype - the type, onset, etc., fields in PhenotypicFeature should be used for this purpose.

7.18.3 assay

An ontology class that describes the assay used to produce the measurement.

EXAMPLE “body temperature” (CMO:0000015) or “Platelets [#/volume] in Blood” (LOINC:26515-7) FHIR mapping: Observation.code.

7.18.4 value

This element represents the result of the measurement. The measurement can be quantitative, such as LOINC:2472-9 (IgM [Mass/volume] in Serum or Plasma), ordinal, or categorical. See 7.36 (Value) and 7.5 (ComplexValue) for more information

7.18.5 time_observed

The time at which the measurement was made. See 7.31 (TimeElement) for more information.

7.18.6 procedure

Clinical procedure performed on the subject to obtain the sample used for the measurement. Examples include blood draw and biopsy. If the procedure can be inferred from the measurement or if the details of the measurement are deemed unimportant (e.g. a blood glucose test is performed on a blood sample obtained with some procedure that is not specified), this element may be omitted. See 7.24 (Procedure) for more information.

7.19 MedicalAction

7.19.1 General

This element describes medications, procedures, other actions taken for clinical management (see Table 25). The element is a list of options.

Table 25 — MedicalAction data model

Field	Type	Multiplicity	Description
action	one of {Procedure | Treatment | RadiationTherapy | TherapeuticRegimen}	1..1	One from a list of medical actions
treatment_target	OntologyClass	0..1	The condition or disease that this treatment was intended to address
treatment_intent	OntologyClass	0..1	The intention of the treatment
response_to_treatment	OntologyClass	0..1	How the patient responded to the treatment
adverse_events	OntologyClass	0..*	Any adverse effect experienced by the patient attributed to the treatment
treatment_termination_reason	OntologyClass	0..1	The reason the treatment was stopped

7.19.2 action

Each MedicalAction element refers to one of the following specific types of medical action: Procedure (7.24), Treatment (7.34), RadiationTherapy (7.26), or TherapeuticRegimen (7.30). See the respective sections for more information.

7.19.3 treatment_target

An OntologyClass to represent the condition or disease that the medical action was intended to address.

7.19.4 treatment_intent

An OntologyClass to represent the intention of the treatment, such as curative, palliative, etc.

7.19.5 response_to_treatment

An OntologyClass to represent how the patient responded to the treatment.

7.19.6 adverse_events

An OntologyClass to represent adverse effects experienced by the patient attributed to the treatment, such as an allergic reaction.

7.19.7 treatment_termination_reason

An OntologyClass to represent the reason why the treatment was stopped (e.g. completed, patient was given a new regimen).

7.20 MetaData

7.20.1 General

This element contains structured descriptions of the resources and ontologies used within the phenopacket (see Table 26). A valid Phenopacket shall have a MetaData element, and application Q/C software should check this. The MetaData element shall have one Resource element for each ontology or terminology whose terms are used in the Phenopacket. For instance, if a MONDO term is used to specify the disease and HPO terms are used to specify the phenotypes of a patient, then the MetaData element shall have one Resource element each for MONDO and HPO.

Table 26 — MetaData data model

Field	Type	Multiplicity	Description
created	A Timestamp	1..1	Representation of the time when the object was created
created_by	string	1..1	Name of person who created the phenopacket
submitted_by	string	0..1	Name of person who submitted the phenopacket
resources	Resource (list)	1..*	A listing of the ontologies/resources referenced in the phenopacket
updates	Update (list)	0..*	List of updates to the phenopacket
phenopacket_schema_version	string	1..1	Schema version of the current phenopacket
external_references	ExternalReference (list)	0..*	A list of external references

EXAMPLE

```
metadata:
  created: "2019-07-21T00:25:54.662Z"
  createdBy: "Peter R."
  resources:
      - id: "hp"
      name: "human phenotype ontology"
      url: "http://purl.obolibrary.org/obo/hp.owl"
      version: "2018-03-08"
      namespacePrefix: "HP"
      iriPrefix: "hp"
      - id: "geno"
      name: "Genotype Ontology"
      url: "http://purl.obolibrary.org/obo/geno.owl"
      version: "19-03-2018"
      namespacePrefix: "GENO"
      iriPrefix: "geno"
      - id: "pubmed"
      name: "PubMed"
      url: "https://www.ncbi.nlm.nih.gov/pubmed/"
      namespacePrefix: "PMID"
  externalReferences:
      - id: "PMID:30808312"
      description: "Bao M, et al. COL6A1 mutation leading to Bethlem myopathy with
recurrent hematuria: a case report. BMC Neurol. 2019;19(1):32."
```

7.20.2 created

This element is an ISO 8601 UTC timestamp for when this phenopacket was created.

EXAMPLE 2018-03-01T00:00:00Z, 2019-04-01T15:10:17.808Z.

7.20.3 created_by

This is a string that represents an identifier for the contributor/ program. The expected syntax and semantics are application dependent.

7.20.4 submitted_by

This is a string that represents an identifier for the person who submitted the phenopacket (who might not be the person who created the phenopacket).

7.20.5 resources

This element contains a listing of the ontologies/resources referenced in the phenopacket. See 7.28 (Resource) for more information.

7.20.6 updates

This element contains a list of Update objects, which contains information about when, what and who updated a phenopacket. This is only necessary when a phenopacket is being used as a persistent record and is being continuously updated. Resources should provide information about how this is being used. See 7.35 (Update) for more information.

7.20.7 phenopacket_schema_version

A string representing the version of the phenopacket-schema according to which a phenopacket was made.

7.20.8 external_references

A list of ExternalReference (7.10), such as the PubMed id of a publication from which a phenopacket was derived.

7.21 OntologyClass

7.21.1 General

This element is used to represent classes (terms) from ontologies and is used in many places throughout the phenopacket standard. It is a simple, two element message that represents the identifier and the label of an ontology class (see Table 27). The ID shall be a CURIE-style identifier. The label should be the corresponding class name. The phenopacket standard requires that the ID and the label match in the original ontology. Occasionally, ontology maintainers change the primary label of a term.

Table 27 — OntologyClass data model

Field	Type	Multiplicity	Description
id	string	1..1	A CURIE-style identifier
label	string	1..1	A human-readable class name

EXAMPLE

```
ontologyClass:
    id: "HP:0001875"
    label: "Neutropenia"
```

7.21.2 id

The ID of an OntologyClass element shall take the form of a CURIE format. It shall reference the namespace prefix of a Resource named in the MetaData so that the class is resolvable.

EXAMPLE HP:0100024, MP:0001284, UBERON:0001690.

7.21.3 label

The human-readable label for the concept. This shall match the ID in the ontology referenced by the namespace prefix in a Resource named in the MetaData.

EXAMPLE Neutropenia.

7.22 Pedigree

7.22.1 General

This element is used to represent a pedigree to describe the family relationships of each sample along with their gender and phenotype (affected status). The information in this element is for use by programs for analysis of a multi-sample VCF file with exome or genome sequences of members of a family, some of whom are affected by a Mendelian disease.

The phenopacket schema has implemented a PED-compatible data-model to promote interoperability between existing PED files and PED software but does not actually store a PED file.

The pedigree is simply a list of Person objects (see Table 28). These objects reflect the elements of a PED file.

Table 28 — Pedigree Data Model

Field	Type	Multiplicity	Description
persons	Person (list)	1..*	List of family members in this pedigree

EXAMPLE

```
pedigree:
 persons:
 - familyId: "family 1"
   individualId: "kindred 1A"
   paternalId: "FATHER"
   maternalId: "MOTHER"
   sex: "MALE"
   affectedStatus: "AFFECTED"
 - familyId: "family 1"
   individualId: "kindred 1B"
   paternalId: "FATHER"
   maternalId: "MOTHER"
   sex: "MALE"
   affectedStatus: "AFFECTED"
 - familyId: "family 1"
   individualId: "MOTHER"
   paternalId: "0"
   maternalId: "0"
   sex: "FEMALE"
   affectedStatus: "UNAFFECTED"
 - familyId: "family 1"
   individualId: "FATHER"
   paternalId: "0"
   maternalId: "0"
   sex: "MALE"
   affectedStatus: "UNAFFECTED"
```

7.22.2 persons

7.22.2.1 General

The persons elements can contain a list of Person, or family members in the pedigree (see Table 29). The Person class represents a row from the PED file indicating the biological parents of the individual, their sex and their AffectedStatus. Person element is equivalent to one row of a PED file.

The family ID and the individual IDs can be made up of letters and digits, and the combination of family and individual ID should uniquely identify each person represented in the PED file. The parents of a person in the pedigree are shown with the corresponding individual IDs. Individuals whose parents are not represented in the PED file are known as founders; their parents are represented by a zero (“0”) in the columns for mother and father.

If a phenopacket is used to represent any of the individuals listed in the Pedigree, the individual_id used in the pedigree shall match the ID of the subject of the phenopacket. The Pedigree may have individuals that do not have an associated phenopacket. This is useful, for instance, if the Pedigree is being used to store the affected/not affected status of family members being examined by exome or genome sequencing. In this case (i.e. where there are no associated phenopackets for the Pedigree individual_id), it is expected that the individual_id fields match the sample identifiers of the exome/genome file.

Table 29 — Person data element

Field	Type	Multiplicity	Description
family_id	string	1..1	Application specific identifier of the family
individual_id	string	1..1	Application specific identifier of the individual/proband
paternal_id	string	1..1	Application specific identifier of the father
maternal_id	string	1..1	Application specific identifier of the mother
sex	Sex	1..1	An enumeration used to represent the sex of an individual
affected_status	AffectedStatus	1..1	An enumeration used to represent the affected status of an individual

7.22.2.2 family_id

A string representing an application specific identifier of the family.

7.22.2.3 individual_id

A string representing an application specific identifier of the individual.

7.22.2.4 paternal_id

A string representing an application specific identifier of the father.

7.22.2.5 maternal_id

A string representing an application specific identifier of the mother.

7.22.2.6 sex

An enumeration of the sex of the individual. See 7.29 (Sex) for more information.

7.22.2.7 affected_status

An enumeration used to represent the affected status of an individual (see Table 30).

In a PED file, affected persons are encoded with “2”, and unaffected persons by “1” (a “0” is used if no information is available). Instead, phenopackets uses an enumeration as shown in Table 23. In a PED file, the sex of individuals is encoded as a “1” for females, “2” for males, and “0” for unknown. Phenopackets uses Sex instead.

Table 30 — AffectedStatus enumerated values

Name	Description
MISSING	It is unknown if the individual has the affected phenotype
UNAFFECTED	The individual does not show the affected phenotype of the proband
AFFECTED	The individual has the affected phenotype of the proband

7.23 PhenotypicFeature

7.23.1 General

This element is intended to be used to describe a phenotype that characterizes the subject of the Phenopacket (see Table 31). For medical use cases the subject will generally be a patient or a proband of a study, and the phenotypes will be abnormalities described by an ontology such as HPO.

Table 31 — PhenotypicFeature data model

Field	Type	Multiplicity	Description
description	string	0..1	Human-readable verbiage not for structured text
type	OntologyClass	1..1	Primary ontology class that represents the phenotype
excluded	boolean	0..1	Flag to indicate whether the phenotype was observed or not
severity	OntologyClass	0..1	Description of the severity of the feature described in type
modifiers	OntologyClass (list)	0..*	One or more ontology terms that are intended to provide more expressive or precise descriptions of a phenotypic feature
onset	TimeElement	0..1	The age or time at which the phenotype feature was first observed
resolution	TimeElement	0..1	The age or time at which the feature resolved or abated
evidence	Evidence (list)	0..*	The evidence for an assertion of the observation of a type

EXAMPLE Recurrent Infantile spasms, which had onset at age 6 months and resoled at age 4 years and 2 months.

```
phenotypicFeature:
    type:
        id: "HP:0012469"
        label: "Infantile spasms"
    modifiers:
        - id: "HP:0031796"
        label: "Recurrent"
    onset:
        age:
            iso8601duration: "P6M"
    resolution:
        age:
            iso8601duration: "P4Y2M"
```

7.23.2 description

This element represents a free-text description of the phenotype. It should not be used as the primary means of describing the phenotype but may be used to supplement the record if ontology terms are not sufficiently able to capture all the nuances.

7.23.3 type

The element represents the primary ontology class which describes the phenotype.

EXAMPLE Craniosynostosis (HP:0001363).

7.23.4 excluded

This element is a flag to indicate whether the phenotype was observed or not. The default is ‘false’, in other words, the phenotype was observed. Therefore, it shall be used in cases to indicate that the phenotype was looked for but found to be absent.

7.23.5 severity

This element is an ontology class that describes the severity of the condition.

EXAMPLE Subclasses of Severity (HP:0012824) or SNOMED:272141005-Severities.

7.23.6 modifiers

This element is a list of ontology class elements that can be empty or contain one or more ontology terms that are intended to provide more expressive or precise descriptions of a phenotypic feature, including attributes such as positionality and external factors that tend to trigger or ameliorate the feature. Concepts can be taken from the hierarchy of clinical modifiers in the HPO (HP:0012823). Please note that severity should be coded in the severity element.

7.23.7 onset

This element can be used to describe the age at which a phenotypic feature was first noticed or diagnosed. For many medical use cases, either the Age sub-element or an ontology class (e.g. from the HPO Onset (HP:0003674) terms) will be used. See 7.31 (TimeElement) for more information.

EXAMPLE HP:0003674 HP:0011462.

7.23.8 resolution

This element can be used to describe the age or time when a phenotypic feature resolved (e.g. disappeared, got better). In the example shown in 7.23.1, infantile spasms no longer occurred after the age of 4 years and 2 months. See 7.31 (TimeElement) for more information.

7.23.9 evidence

This field contains one or more Evidence elements that specify how the phenotype was determined. See 7.9 (Evidence) for more detail.

7.24 Procedure

7.24.1 General

The Procedure element represents a clinical procedure performed on a subject (see Table 32). If the Procedure element is used, it shall contain a code element.

Table 32 — Procedure data model

Field	Type	Multiplicity	Description
code	OntologyClass	1..1	The clinical procedure performed on a subject
body_site	OntologyClass	0..*	The specific body site where the procedure was performed
performed	TimeElement	0..*	The age or time when the procedure was performed

EXAMPLE A skin biopsy from the skin of the forearm is performed on an individual who is 25 years old.

```
procedure:
    code:
        id: "NCIT:C28743"
        label: "Punch Biopsy"
    bodySite:
        id: "UBERON:0003403"
        label: "skin of forearm"
    performed:
        age:
            iso8601duration: "P25Y"
```

7.24.2 code

This element is an OntologyClass that represents clinical procedure performed on a subject.

EXAMPLE

```
code:
   id: "NCIT:C51585"
   label: "Biopsy of Soft Palate"
```

7.24.3 body site

In cases where it is not possible to represent the procedure adequately with a single OntologyClass, the body site should be indicated using a separate ontology class.

EXAMPLE

```
code:
    id: "NCIT:C28743"
    label: "Punch Biopsy"
 bodySite:
    id: "UBERON:0003403"
    label: "skin of forearm"
```

7.24.4 performed

The age or time when the procedure was performed. See 7.31 (TimeElement) for more detail.

7.25 Quantity

7.25.1 General

This element is meant to denote quantities of items such as medications (see Table 33). The unit of a dose can be expressed with NCIT terms such as Milligram, Microgram, or Unit. The value should be expressed as a number.

Table 33 — Quantity data model

Field	Type	Multiplicity	Description
unit	OntologyClass	1..1	The unit
value	double	1..1	The value of the quantity in the appropriate units
reference_range	ReferenceRange	0..1	The normal range for the value

EXAMPLE 1 The following message could be used to represent the quantity corresponding to a 15 mg tablet of Meloxicam.

```
unit:
  id: "NCIT:C28253"
  label: "Milligram"
value: 15.0
```

EXAMPLE 2 The following message could be used to represent the quantity corresponding to a bolus of 5 000 units of Heparin.

```
unit:
  id: "NCIT:C44278"
  label: "Unit"
value: 5000
```

EXAMPLE 3 The following message shows a quantity for a platelet count per microliter, with a reference range.

```
unit:
  id: "UO:0000316"
  label: "cells per microliter"
value: 300000.0
referenceRange:
  unit:
    id: "UO:0000316"
    label: "cells per microliter"
  low: 150000.0
  high: 450000.0
```

7.25.2 unit

An ontology term for the unit measured. Ontology terms for units can be taken from the NCIT subhierarchy for Unit of Measure (Code C25709).

7.25.3 value

The corresponding value of the quantity in the indicated units.

7.25.4 reference_range

The expected normal range of the value measured. See 7.27 (ReferenceRange) for more information.

7.26 RadiationTherapy

7.26.1 General

Radiation therapy (or radiotherapy) uses ionizing radiation, generally as part of cancer treatment to control or kill malignant cells. This element is contained within a MedicalAction and describes the radiation therapy administered (see Table 34).

Table 34 — RadiationTherapy data model

Field	Type	Multiplicity	Description
modality	OntologyClass	1..1	The modality of radiation therapy
body_site	OntologyClass	1..1	The anatomical site where radiation therapy was administered
dosage	int32	1..1	The total dose given in units of Gray (Gy)
fractions	int32	1..1	The total number of fractions delivered as part of treatment

7.26.2 modality

An ontology class to represent the modality of radiation therapy. Terms from the NCIT should be used. The NCIT terms themselves specify whether the radiation therapy was administered externally or internally. For instance, NCIT terms from the Internal Radiation Therapy (NCIT:C15195) subhierarchy would be used to indicate a type of internal radiation therapy.

EXAMPLE Electron Beam (NCIT:C28039), Proton Radiation (NCIT:C40431), Photon Beam Radiation Therapy (NCIT:C104914), High-LET Heavy Ion Therapy (NCIT:C15458).

7.26.3 body_site

The anatomical site that was irradiated. An UBERON term can be used for this field.

7.26.4 dosage

The total dosage administered, indicated in units of Gray.

7.26.5 fractions

The number of fractions into which the radiation dosage was divided.

7.27 ReferenceRange

7.27.1 General

This element is provided to support the Measurement element, which can be used to report a numerical value such as Platelets [#/volume] in Blood (LOINC:26515-7). See Table 35 for the fields in this element.

Table 35 — ReferenceRange data model

Field	Type	Multiplicity	Description
unit	OntologyClass	1..1	The reference unit
low	double	1..1	The low range of normal
high	double	1..1	The upper range of normal

EXAMPLE There are several ontologies that provide terms for units of measurement, including the Units of measurement ontology and the NCIT, from the subhierarchy for Unit of Measure (Code C25709). The following example shows the reference range for platelet count per microliter. The normal range for circulating platelets is 150 000 to 450 000 platelets per microliter.

```
referenceRange:
  unit:
      id: "UO:0000316"
      label: "cells per microliter"
  low: 150000.0
  high: 450000.0
```

7.27.2 unit

An ontology term for the reference range of the unit (being measured in the Measurement element).

7.27.3 low

The lower range of normal for the reference range.

7.27.4 high

The upper range of normal for the reference range.

7.28 Resource

7.28.1 General

The Resource element is a description of an external resource used for referencing an object (see Table 36). The resource can be an ontology such as the HPO or SNOMED or another data resource such as the HGNC or ClinVar. For an ontology, the url shall point to the obo or owl file. This information can also be found at the EBI Ontology Lookup Service.

A Resource is used to contain data used to expand CURIE identifiers when used in an id field. This is known as identifier resolution.

The MetaData element uses one resource element to describe each resource that is referenced in the phenopacket.

Table 36 — Resource data model

Field	Type	Multiplicity	Description
id	string	1..1	The resource identifier
name	string	1..1	The formal name of the resource or ontology referred to by the id element
url	string	1..1	Uniform resource locator of the resource
version	string	1..1	The version of the resource or ontology used to make the annotation
namespace_prefix	string	1..1	The namespace prefix of the resource
iri_prefix	string	1..1	The Internationalized Resource Identifier

EXAMPLE

```
resource:
  id: "hp"
  name: "Human Phenotype Ontology"
  url: "http://www.human-phenotype-ontology.org"
  version: "2018-03-08"
  namespacePrefix: "HP"
  iriPrefix: "http://purl.obolibrary.org/obo/HP_"
```

Non-ontology resources which use CURIEs as their native identifiers should be treated in a similarly resolvable manner.

```
resource:
  id: "hgnc"
  name: "HUGO Gene Nomenclature Committee"
  url: "https://www.genenames.org"
  version: "2019-08-08"
  namespacePrefix: "HGNC"
  iriPrefix: "https://www.genenames.org/data/gene-symbol-report/#!/hgnc_id/"
```

Using this Resource definition, it is possible for software to resolve the identifier HGNC:12805 to https://www.genenames.org/data/gene-symbol-report/#!/hgnc_id/12805.

Non-ontology resources that do not use CURIEs as their native identifiers shall use the namespace from identifiers.org, when present.

```
resource:
  id: "uniprot"
  name: "UniProt Knowledgebase"
  url: "https://www.uniprot.org"
  version: "2019_07"
  namespacePrefix: "uniprot"
  iriPrefix: "https://purl.uniprot.org/uniprot/"
```

Using this Resource definition, it is possible for software to resolve the identifier uniprot:Q8H0D3 to https://purl.uniprot.org/uniprot/Q8H0D3.

7.28.2 id

For OBO ontologies, the value of this string shall always be the official OBO ID, which is always equivalent to the ID prefix in lower case. See http://obofoundry.org for a complete list.

EXAMPLE hp, go, mp, mondo.

For other resources that do not use native CURIE identifiers (e.g. SNOMED, UniProt, ClinVar), use the prefix in identifiers.org.

7.28.3 name

The name of the ontology referred to by the id field. For OBO Ontologies, the value of this string should be the same as the title field on http://obofoundry.org. Other resources should use the official title for that resource. This field is purely for information purposes and software should not encode any assumptions.

EXAMPLE Human Phenotype Ontology.

7.28.4 url

The URL of the resource. For OBO ontologies, this shall be the PURL.

EXAMPLE http://purl.obolibrary.org/obo/hp.owl, http://purl.obolibrary.org/obo/hp.obo.

Other resources should link to the official or top-level URL.

EXAMPLE https://www.uniprot.org, https://www.genenames.org.

7.28.5 version

The version of the resource or ontology used to make the annotation. For OBO ontologies, this shall be the versionIRI. For other resources this should be the native version of the resource. For resources without release versions, this field should be left blank.

EXAMPLE 2018-03-08, UniProt - "2019_08", DbSNP - "153".

7.28.6 namespace_prefix

The prefix used in the CURIE of an OntologyClass. An HPO term will have a CURIE like HP:0012828 that should be used in combination with the iri_prefix to form a fully resolvable IRI.

EXAMPLE HP, MP, ECO, uniprot.

7.28.7 iriPrefix

The full IRI prefix, which can be used with the namespace_prefix, and the OntologyClass id to resolve to an IRI for a term. Tools such as the curie-util (https://github.com/prefixcommons/curie-util) can utilize this to produce fully resolvable IRIs for an OntologyClass.

EXAMPLE https://purl.uniprot.org/uniprot/.

7.28.8 CURIE

The CURIE is defined by the W3C as a means of encoding a "Compact URI". It is a simple string taking the form of colon (:) separated prefix and reference elements - prefix:reference.

EXAMPLE HP:0012828, HGNC:12805.

CURIE identifiers should be used where possible.

Not all resources use CURIEs as identifiers (e.g. SNOMED, UniProt, ClinVar, PubMed). In these cases, it is often possible to create a CURIE form of an identifier by using the prefix for that resource from identifiers.org.

Where no CURIE prefix is present in identifiers.org, it is possible for a Resource to define a locally scoped namespace, although if a phenopacket is being shared publicly this is not recommended if the resource is not publicly resolvable.

When using a CURIE identifier, a corresponding Resource shall also be included in the MetaData section.

7.28.9 Identifier resolution

A CURIE identifier can be resolved to an external resource using the Resource element by looking-up the CURIE prefix against the Resource::namespacePrefix and then appending the CURIE reference to the Resource::iriPrefix.

EXAMPLE Software can use the HPO Resource shown above to resolve the following HPO term encoding the concept of Severe:

```
ontologyClass:
  id: "HP:0012828"
  label: "Severe"
```

The id HP:0012828 can be split into the prefix - ‘HP’ and reference - ‘0012828’. The ‘HP’ prefix matches the Resource::namespacePrefix so we can append the reference ‘0012828’ to the Resource::iriPrefix: which produces the URI http://purl.obolibrary.org/obo/HP_0012828. The term can be resolved to http://purl.obolibrary.org/obo/HP_0012828.

7.29 Sex

An enumeration used to represent the sex of an individual (see Table 37). This element does not represent gender identity or karyotypic sex, but instead represents typical “phenotypic sex”, as would be determined by a midwife or physician at birth. This is an enumerated type, therefore the values represented below are the only legal values. The value of this type shall not be null, instead it shall use the 0 (zero) ordinal element as the default value, should none be specified.

Table 37 — Sex element enumerated types

Name	Ordinal	Description
UNKNOWN_SEX	0	Not assessed or not available. Maps to NCIT:C17998
FEMALE	1	Female sex. Maps to NCIT:C46113
MALE	2	Male sex. Maps to NCIT:C46112
OTHER_SEX	3	It is not possible to accurately assess the applicability of MALE/FEMALE. Maps to NCIT:C45908

EXAMPLE

```
sex: "UNKNOWN_SEX"
```

7.30 TherapeuticRegimen

7.30.1 General

This element represents a therapeutic regimen, which will involve a specified set of treatments for a particular condition (see Table 38). It can be thought of as a shorthand for a more specific set of treatments. This element is supposed to reference a more detailed regimen specification.

Table 38 — TherapeuticRegimen data model

Field	Type	Multiplicity	Description
identifier	one of {OntologyClass | ExternalReference}	1..1	An identifier of the regimen
start_time	TimeElement	0..1	The time when the regimen was started
end_time	TimeElement	0..1	The time when the regimen ended
status	RegimenStatus	1..1	The current status of the regimen for the enclosing MedicalAction on the Individual

EXAMPLE The following example describes twice daily dosing of 30 mg of losartan given orally.

```
therapeuticRegimenTreatment:
  externalReference:
    id: "NCT04576091"
    reference: "https://clinicaltrials.gov/ct2/show/NCT04576091"
    description: "Testing the Addition of an Anti-cancer Drug, BAY1895344, With Radiation\
      \ Therapy to the Usual Pembrolizumab Treatment for Recurrent Head and Neck Cancer"
  startTime:
    timestamp: "2020-03-15T13:00:00Z"
  regimenStatus: "STARTED"
```

7.30.2 identifier

An OntologyClass (7.21) or ExternalReference (7.10) representing the therapeutic regimen that the subject (Individual) has followed.

7.30.3 start_time

When the regimen was started, as represented by a TimeElement (7.31).

7.30.4 end_time

When the regimen ended, as represented by a Time Element (7.31). An empty end_time with a populated start_time would indicate that the regimen was ongoing.

7.30.5 regimen_status

An enumeration representing the status of the regimen – whether it has started, completed or was discontinued (see Table 39). Regimens that were discontinued should record any adverse events (MedicalAction.adverse_events) and the reason for termination (MedicalAction.treatment_termination_reason) in the enclosing MedicalAction message.

Table 39 — RegimenStatus enumeration values

Name	Ordinal	Description
UNKNOWN_STATUS	0	The status of the regimen is unknown
STARTED	1	The regimen was started
COMPLETED	2	The regimen was completed
DISCONTINUED	3	The regimen was discontinued

7.31 TimeElement

7.31.1 General

This element intends to bundle all the various ways of denoting time or age in the phenopackets schema, to ensure that all references to time and age throughout the phenopacket standard are uniform. A TimeElement shall include one of the options in the data model (see Table 40).

Table 40 — TimeElement data model

Field	Type	Multiplicity	Description
gestational_age	GestationalAge	(one of the options)	A measure of the age of a pregnancy
age	Age	(one of the options)	Represents age as an ISO 8601 duration
age_range	AgeRange	(one of the options)	Indicates that the individual’s age lies within a given range

Table 40 *(continued)*

Field	Type	Multiplicity	Description
ontology_class	OntologyClass	(one of the options)	Indicates the age of the individual as an ontology class
timestamp	Timestamp	(one of the options)	Indicates a specific time
interval	TimeInterval	(one of the options)	Indicates an interval of time

EXAMPLE The following example shows a TimeElement with the Age option.

```
timeElement:
    age:
        iso8601duration: "P25Y"
```

7.31.2 gestational_age

A measure of the age of a pregnancy. Gestation, defined as the time between conception and birth, is measured in weeks and days from the first day of the last menstrual period. See 7.14 (GestationalAge) for more information.

7.31.3 age

This element can be used to represent age as an ISO 8601 duration. See 7.2 (Age) for more information.

EXAMPLE P40Y10M05D.

7.31.4 age_range

This element can be used to indicate that the individual's age lies within a given range, which may be desirable to help preserve privacy. See 7.3 (AgeRange) for more information.

7.31.5 ontology_class

If an OntologyClass is used to represent the age of onset of a phenotypic feature, then terms for age of onset can be chosen from the Onset subhierarchy of the HPO. See 7.21 (OntologyClass) for more information.

7.31.6 timestamp

A Timestamp can be used to represent a specific time. All timestamps in a phenopacket can be shifted by the same amount to help preserve privacy if desired. See 7.32 (TimeInterval) for more information.

7.31.7 interval

This element can be used to represent a specific interval of time. See 7.32 (TimeInterval) for more information.

7.32 TimeInterval

7.32.1 General

A time interval is meant to denote an interval of time whose beginning and end are defined by a timestamp (see Table 41). See 7.33 (Timestamp) for more information.

Table 41 — TimeInterval data model

Field	Type	Multiplicity	Description
start	Timestamp	1..1	The start of the interval
end	Timestamp	1..1	The end of the interval

EXAMPLE The following message can be used to represent the interval from March 15, 2020, 1PM to March 25, 2020, 9PM.

```
timeInterval:
  start: "2020-03-15T13:00:00Z"
  end: "2020-03-25T09:00:00Z"
```

In some cases, it may be desirable to shift all specific dates in a phenopacket by the same random amount. For instance, all dates can be shifted by 2 years. In this case the above interval element would be represented as follows:

```
timeInterval:
  start: "2018-03-15T13:00:00Z"
  end: "2018-03-25T09:00:00Z"
```

7.32.2 start

The date and time of the start of the interval.

7.32.3 end

The date and time of the end of the interval.

7.33 Timestamp

In phenopackets, Timestamp is defined as a date time string in accordance with the ISO 8601 series, i.e. ISO 8601 Timestamp. The following text paraphrases the description of how this is represented in protobuf as JSON Timestamp.

The format for this is "{year}-{month}-{day}T{hour}:{min}:{sec}[.{frac_sec}]Z" where {year} is always expressed using four digits while {month}, {day}, {hour}, {min}, and {sec} are zero-padded to two digits each. The fractional seconds, which can go up to 9 digits (i.e. up to 1 nanosecond resolution), are optional. The "Z" suffix indicates the timezone ("UTC"); the timezone is required.

EXAMPLE "2021-06-02T16:52:15.01Z" encodes 15,01 seconds past 16:52 UTC on June 2, 2021.

In JavaScript, one can convert a date object to this format using the standard toISOString() method. In Python, a standard datetime.datetime object can be converted into this format using strftime with the time format spec '%Y-%m-%dT%H:%M:%S.%fZ'. Likewise, in Java, one can use the DateTimeFormatter.ISO_DATE_TIME to obtain a formatter capable of generating timestamps in this format.

7.34 Treatment

7.34.1 General

This element represents treatment with an agent such as a drug (pharmaceutical agent), broadly defined as prescription and over-the-counter medicines, vaccines, and large-molecule biologic therapies (see Table 42).

Table 42 — Treatment data model

Field	Type	Multiplicity	Description
agent	OntologyClass	1..1	The drug or therapeutic agent
route_of_administration	OntologyClass	0..1	How the drug was administered
dose_intervals	DoseInterval (list)	0..*	The doses of agent administered over a defined interval
drug_type	DrugType	0..1	Context of the drug administration
cumulative_dose	Quantity	0..1	The cumulative dose administered during the period of the treatment

EXAMPLE 1 The following example describes twice daily dosing of 30 mg of losartan given orally.

```
treatment:
    agent:
        id: "DrugCentral:1610"
        label: "losartan"
    routeOfAdministration:
        id: "NCIT:C38288"
        label: "Oral Route of Administration"
    doseIntervals:
        - quantity:
            unit:
                id: "UO:0000022"
                label: "milligram"
            value: 30.0
    scheduleFrequency:
        id: "NCIT:C64496"
        label: "Twice Daily"
    interval:
        start: "2020-03-15T13:00:00Z"
        end: "2020-03-25T09:00:00Z"
    drugType: "PRESCRIPTION"
```

EXAMPLE 2 The following example specifies that aclarubicin (a type of anthracycline) was given intravenously every three weeks in the time period from 2020-07-10 to 2020-08-10, as part of a cancer chemotherapy treatment for a cumulative dose of 200 mg/kg.

```
    treatment:
        agent:
            id: "DrugCentral:80"
            label: "aclarubicin"
        routeOfAdministration:
            id: "NCIT:C38276"
            label: "Intravenous Route of Administration"
        doseIntervals:
            - quantity:
                unit:
                    id: "NCIT:C124458"
                    label: "Milligram per Kilogram per Dose"
                value: 100.0
            scheduleFrequency:
                id: "NCIT:C64535"
                label: "Every Three Weeks"
            interval:
                start: "2020-07-10T00:00:00Z"
                end: "2020-08-10T00:00:00Z"
        drugType: "EHR_MEDICATION_LIST"
    cumulativeDose:
        unit:
            id: "EFO:0002902"
            label: "milligram per kilogram"
        value: 200.0
```

7.34.2 agent

An ontology term representing the therapeutic agent. This can be a term from DrugCentral, RxNorm, Drugbank, ChEBI, or other ontologies.

7.34.3 route_of_administration

How the drug is administered, e.g. by mouth or intravenously. This can be specified by ontology terms from the NCIT subhierarchy for Route of Administration.

7.34.4 dose_intervals

A block of time in which the dosage of a medication was constant, for example, 30 mg/day for an interval of 10 days. See 7.7 (DoseInterval) for more information.

7.34.5 drug_type

The context in which a drug was administered. See 7.8 (DrugType) for more information.

7.34.6 cumulative_dose

The cumulative dose, defined as the total dose from repeated exposures to chemotherapy, monitoring of which is an important part of treatment with chemotherapy. For instance, cardiac side effect risk increases with greater cumulative doses of anthracycline.

7.35 Update

7.35.1 General

Update is a class for holding data about an update event to a record (see Table 43). It is used to hold brief details about when it occurred and optionally, who or what made the update and any pertinent information regarding the content and/or reason for the update. The class is used in the MetaData element.

Table 43 — Update data model

Field	Type	Multiplicity	Description
timestamp	ISO 8601 UTC Timestamp	1..1	ISO 8601 UTC timestamp at which the record was updated
updated_by	string	0..1	Information about the person/organization/network that has updated the phenopacket
comment	string	0..1	Textual comment about the changes made to the content and/or the reason for the update

EXAMPLE

```
update:
  timestamp: "2018-06-10T10:59:06Z"
```

Optionally, with extra information:

```
update:
  timestamp: "2018-06-10T10:59:06Z"
  updatedBy: "Julius J."
  comment: "added phenotypic features to individual patient:1"
```

7.35.2 timestamp

An ISO 8601 UTC timestamp for when this phenopacket was updated. See 7.33 (Timestamp) for more information.

7.35.3 updated_by

Information about the person/organisation/network that has updated the phenopacket.

7.35.4 comment

A textual comment about the changes made to the content and/or reason for the update. This should be a brief description only; it is not the actual update.

7.36 Value

7.36.1 General

The value element is meant to be used as part of the Measurement element, and it represents the outcome of a measurement (see Table 44).

Table 44 — Value data model

Field	Type	Multiplicity	Description
value	one of {Quantity | OntologyClass}	1..1	The outcome (value) of a measurement

EXAMPLE The following example shows a Value used for quantitative measurement.

```
value:
    quantity:
        unit:
            id: "UO:0000316"
            label: "cells per microliter"
        value: 24000.0
        referenceRange:
            unit:
                id: "UO:0000316"
                label: "cells per microliter"
            low: 150000.0
            high: 450000.0
```

The following example shows a Value used for an ordinal measurement.

```
value:
    ontologyClass:
        id: "NCIT:C25626"
        label: "Present"
```

7.36.2 value

The outcome of a measurement. See 7.25 (Quantity) or 7.21 (OntologyClass) for more information. For ordinal measurements, the following terms can be useful: Present (NCIT:C25626), Absent (NCIT:C48190), Abnormal (NCIT:C25401), Elevated (NCIT:C25493), Reduced (NCIT:C25640).

7.37 VariationDescriptor

7.37.1 General

Variation descriptors are part of the VRSATILE framework[9], a set of conventions extending the GA4GH VRS[10]. Descriptors allow for the complementary use of human-readable labels, descriptions, alternate contexts, and identifier cross-references alongside the precise computable representation of variation concepts provided by VRS.

Consequently, many forms and formats of variation can be used in variation descriptors, including HGVS descriptions, VCF records, and SPDI alleles. VRS variation objects for representing variants should be used when possible.

The VariationDescriptor element should be used to describe candidate variants or diagnosed causative variants (see Table 45). The VariationDescriptor element itself is an element of a VariantInterpretation (see 7.38). If it is present, the phenopacket standard has the following requirements.

A variation may refer to an external source, for example the ClinGen allele registry, ClinVar, dbSNP, dbVAR, etc. using the id field. A CURIE identifier and corresponding Resource should be used. When indicating multiple alternate ids for a variation, use the alternate_ids field. Multiple alleles in-cis can be modeled as a VRS Haplotype.

The zygosity of the variant as determined in all the samples represented in this phenopacket is represented using a list of terms taken from the GENO. For instance, if a variant affects one of two alleles at a certain locus, the zygosity can be recorded using the term heterozygous (GENO:0000135). This value is stored in the VariationDescriptor alleleic_state field.

Table 45 — VariationDescriptor data model

Field	Type	Multiplicity	Description
id	string	1..1	A descriptor id that is unique within the phenopacket
variation	Variation	0..1	The VRS variation object
label	string	0..1	A primary label for the variation
description	Variation	0..1	A free-text description of the variation
gene_context	GeneDescriptor	0..1	A specific gene context that applies to this variant
expressions	Expression	0..*	HGVS, SPDI, and gnomAD-style strings should be represented as Expressions
vcf_record	VcfRecord	0..1	A VCF record of the variant. This should be a single allele, the VCF genotype (GT) field should be represented in the allelic_state
xrefs	string	0..*	List of CURIEs representing associated concepts
alternate_labels	string	0..*	Common aliases for a variant
extensions	Extension	0..*	List of resource-specific extensions needed to describe the variation
molecule_context	MoleculeContext	1..1	The molecular context of the vrs variation
structural_type	OntologyClass	0..1	The structural variant type associated with this variant, such as a substitution, deletion or fusion
vrs_ref_allele_seq	string	0..1	A sequence corresponding to a “ref allele”, describing the sequence expected at a SequenceLocation reference
allelic_state	OntologyClass	0..1	The zygosity of the variant as determined in all the samples represented in this phenopacket

EXAMPLE These examples will show how the ClinVar allele 13294 (https://www.ncbi.nlm.nih.gov/clinvar/variation/13294/) can be represented using a VariationDescriptor. While it is possible to combine all these in a single message, they have been separated for clarity.

Below is an example that represents the genomic variation using VRS, however, VRS is capable of representing the variation in genomic, transcript or protein coordinates.

```
variationDescriptor:
  id: "clinvar:13294"
  variation:
    allele:
      sequenceLocation:
        sequenceId: "NC_000010.11"
        sequenceInterval:
          startNumber:
            value: "121496700"
          endNumber:
            value: "121496701"
      literalSequenceExpression:
        sequence: "G"
  moleculeContext: "genomic"
  vrsRefAlleleSeq: "T"
  allelicState:
    id: "GENO:0000135"
    label: "heterozygous"
```

Variants can be represented using the HGVS nomenclature as follows. For example, the HGVS expression NM_000226.3:c.470T>G indicates that a T at position 470 of the sequence represented by version 3 of NM_000226 (which is the mRNA of the human keratin 9 gene KRT9). It is recommended to use a tool such as VariantValidator or Mutalyzer to validate the HGVS string. See the HGVS recommendations for details about the HGVS nomenclature.

```
variationDescriptor:
  id: "clinvar:13294"
  expressions:
  - syntax: "hgvs"
    value: "NM_000226.3:c.470T>G"
  allelicState:
    id: "GENO:0000135"
    label: "heterozygous"
```

Below is an example the represents the genomic variation using VCF.

```
variationDescriptor:
    id: "clinvar:13294"
    vcfRecord:
        genomeAssembly: "GRCh38"
        chrom: "10"
        pos: 121496701
        id: "rs121918506"
        ref: "T"
        alt: "G"
        qual: "."
        filter: "."
        info: "."
    zygosity:
        id: "GENO:0000135"
        label: "heterozygous"
```

The SPDI notation is a notation that uses the same normalisation protocol as VRS. Users should familiarize themselves with this notation, which differs in important ways from other standards such as VCF and HGVS. Tools for interconversion between SPDI, HGVS and VCF exist at the NCBI.

The SPDI notation represents variation as deletion of a sequence (D) at a given position (P) in reference sequence (S) followed by insertion of a replacement sequence (I) at that same position. Position 0 indicates a deletion that starts immediately before the first nucleotide, and position 1 represents a deletion interval that starts between the first and second residues, etc. Either the deleted or the inserted interval may be empty, resulting in a pure insertion or deletion. The deleted and inserted sequences in SPDI are all written on the positive strand for two-stranded molecules.

```
variationDescriptor:
  id: "clinvar:13294"
  expressions:
  - syntax: "spdi"
    value: "NC_000010.11:121496700:T:G"
  allelicState:
    id: "GENO:0000135"
    label: "heterozygous"
```

The ISCN[11] includes band names, symbols and abbreviated terms used in the description of human chromosome and chromosome abnormalities. For example, del(6)(q23q24) describes a deletion from band q23 to q24 on chromosome 6.

```
variationDescriptor:
  id: "id:A"
  expressions:
  - syntax: "iscn"
    value: "t(8;9;11)(q12;p24;p12)"
```

7.37.2 id

A descriptor ID that shall be unique within the phenopacket. A variation may refer to an external source, for example the ClinGen allele registry, ClinVar, dbSNP, dbVAR, etc. using the id field.

7.37.3 variation

The VRS Variation Object[10]. VRS is a GA4GH standard which provides a computable representation of variation, be it a genomic, transcript or protein variation. VRS also provides mechanisms for representing haplotypes and systemic variation such as CNVs[10].

7.37.4 label

An optional primary label for the variation.

7.37.5 description

An optional free-text description of the variation.

7.37.6 gene_context

A specific gene context that applies to this variant. See 7.12 (GeneDescriptor) for more information.

7.37.7 expression

The Expression class is designed to enable descriptions based on a specified nomenclature or syntax for representing an object (see Table 46). Common examples of expressions for the description of molecular variation include the HGVS and ISCN nomenclatures.

It is recommended to use one of the following values in the syntax field: hgvs, iscn, spdi.

Table 46 — Expression data model

Field	Type	Multiplicity	Description
syntax	string	1..1	A name for the expression syntax
value	string	1..1	The concept expression as a string
version	string	0..1	An optional version of the expression syntax

7.37.8 vcf_record

This element is used to describe variants using VCF, which is in near universal use for exome, genome, and other next-generation-sequencing-based variant calling (see Table 47). It is an appropriate option to use for variants reported according to their chromosomal location as derived from a VCF file.

In the phenopacket format, it is expected that one VcfRecord message describes a single allele (in contrast to the actual VCF format that allows multiple alleles at the same position to be reported on the same line; to report these in phenopacket format, two VariantDescriptor messages would be required). In general, the VcfRecord should be used only for the purposes of reporting variants of specific interest, such as in the VariantInterpretation, for cases requiring larger numbers of variants in VCF format, the File should be used.

For structural variation, the info field should contain the relevant information. In general, the info field should only be used to report structural variants and it is not expected that the phenopacket will report the contents of the info field for single nucleotide and other small variants.

Table 47 — VcfRecord data model

Field	Type	Multiplicity	Description
genome_assembly	string	1..1	Identifier for the genome assembly used to call the allele
chrom	string	1..1	Chromosome or contig identifier
pos	int	1..1	The reference position, with the 1st base having position 1
id	string	0..1	An identifier, using a semicolon-separated list of unique identifiers where available. If this is a dbSNP variants the number(s) should be used
ref	string	1..1	The reference base
alt	string	1..1	The alternate base
qual	string	0..1	The quality, or Phred-scaled quality score for the assertion made in the alt field
filter	string	0..1	The filter status. “PASS” if this position has passed all filters
info	string	0..1	Additional information, represented using a semicolon-separated series of additional information fields

7.37.9 xrefs

A list of CURIEs representing associated concepts. Allele registry, ClinVar, or other related IDs should be included as xrefs.

7.37.10 alternate_labels

The common aliases for a variant.

EXAMPLE EGFR vIII is an alternate label.

7.37.11 extensions

The Extension class provides a means to extend descriptions with other attributes unique to a content provider (see Table 48). These extensions are not expected to be natively understood by all users but may be used for pre-negotiated exchange of message attributes when needed.

Table 48 — Expression Data Model

Field	Type	Multiplicity	Description
name	string	1..1	A name for the Extension
value	string	1..1	A string representation of a user defined object

7.37.12 molecule_context

The molecular context of the variant. The default is unspecified_molecule_context.

7.37.13 structural_type

The structural variant type associated with this variant, such as a substitution, deletion, or fusion. A descendent term of SO:0001537 should be used.

7.37.14vrs_ref_allele_seq

A sequence corresponding to a “ref allele”, describing the sequence expected at a SequenceLocation reference.

7.37.15allelic_state

The zygosity of the variant as determined in all of the samples represented in a phenopacket using a list of terms taken from the GENO. For instance, if a variant affects one of two alleles at a certain locus, the zygosity could be recorded using the term heterozygous (GENO:0000135).

7.38 VariantInterpretation

7.38.1 General

This element represents the interpretation of a variant according to the ACMG variant interpretation guidelines (see Table 49).

Table 49 — VariantInterpretation data model

Field	Type	Multiplicity	Description
acmg_pathogenicity_classification	AcmgPathogenicityClassification	1..1	One of the five ACMG pathogenicity categories, default is UNCERTAIN_SIGNIFICANCE
therapeutic_actionability	TherapeuticActionability	1..1	The therapeutic actionability of the variant, default is UNKNOWN_ACTIONABILITY
variant	VariantDescriptor	1..1	A genetic/genomic variant

EXAMPLE The following element shows how to denote an interpretation of a variant as pathogenic.

```
variantInterpretation:
  acmgPathogenicityClassification: "PATHOGENIC"
  variationDescriptor:
    expressions:
    - syntax: "hgvs"
      value: "NM_001848.2:c.877G>A"
    allelicState:
      id: "GENO:0000135"
      label: "heterozygous"
```

7.38.2 acmg_pathogenicity_classification

An enumeration of the ACMG recommended five-tier classification system[12] (see Table 50).

Table 50 — AcmgPathogenicityClassification enumeration values

Name	Ordinal	Description
NOT_PROVIDED	0	The variant has not been subject to classification
BENIGN	1	This variant does not cause disease
LIKELY_BENIGN	2	This variant is not expected to have a major effect on disease. However, the scientific evidence is currently insufficient to prove this conclusively
UNCERTAIN_SIGNIFICANCE	3	There is not enough information to support a more definitive classification of this variant
LIKELY_PATHOGENIC	4	There is a high likelihood (greater than 90% certainty) that this variant is disease-causing
PATHOGENIC	5	This variant directly contributes to the development of disease

7.38.3 therapeutic_actionability

An enumeration flagging the variant as being a candidate for treatment/ clinical intervention of the disorder caused by this variant, which could improve the clinical outcome (see Table 51).

Table 51 — TherapeuticActionability enumeration values

Name	Ordinal	Description
UNKNOWN_ACTIONABILITY	0	There is not enough information to support any therapeutic actionability for this variant
NOT_ACTIONABLE	1	This variant has no therapeutic actionability
ACTIONABLE	2	This variant is known to be therapeutically actionable

7.38.4 variant

The subject of the variant interpretation. See 7.37 (VariationDescriptor) for more information.

7.39 VitalStatus

7.39.1 General

This element can be used to report whether the individual is living or dead at the timepoint when the phenopacket was created (or if the status is unknown) (see Table 52). In practice, this element is useful in cohort studies in which the association of some treatment or genetic variation is compared with mortality. For many other applications, it may not be necessary to use a VitalStatus element.

Table 52 — VitalStatus data model

Field	Type	Multiplicity	Description
status	Status	1..1	The status of the individual, whether alive, deceased, or unknown
time_of_death	TimeElement	0..1	The time of death of the individual
cause_of_death	OntologyClass	0..1	The cause of death of the individual
survival_time_in_days	integer	0..1	The number of days the patient was alive after their primary diagnosis

EXAMPLE

```
vitalStatus:
    status: "DECEASED"
    timeOfDeath:
        timestamp: "2015-10-01T10:54:20.021Z"
    causeOfDeath:
        id: "NCIT:C36263"
        label: "Metastatic Malignant Neoplasm"
```

The example below shows how to report the individual is alive.

```
vitalStatus:
    status: "ALIVE"
```

7.39.2 status

An enumeration of the status of the individual (see Table 53).

Table 53 — Status enumeration values

Name	Ordinal	Description
UNKNOWN_STATUS	0	Not assessed or not available
ALIVE	1	The individual is alive. Maps to NCIT:C37987
DECEASED	2	The individual is deceased. Maps to NCIT:C28554

7.39.3 time_of_death

The time of death of the individual. It should be left blank if patient is not known to be deceased. See 7.31 (TimeElement) for more information.

7.39.4 cause_of_death

An ontology class representing the cause of death of the individual. It should be left blank if patient is not known to be deceased.

7.39.5 survival_time_in_days

The number of days the patient was alive after their primary diagnosis.

Annex A
(informative)

Working with Phenopackets

A.1 General

While it is possible to inter-operate with other services using JSON produced from hand-crafted/alternative implementations, it is recommended to use the schema to compile any required language implementations. This Annex provides several examples that demonstrate how to work with Phenopackets in Java and C++. There are also Python examples in the source code test directory. All three language implementations are automatically produced as part of the build (Java Build).

NOTE These examples show version 1.0 of Phenopackets.

A.2 Working with Phenopackets in Java

A.2.1 The Java Code

Below is Java code that demonstrates the basic methodology for building a Phenopacket. The entire code was put into one function for didactic purposes, but real-life code can be more structured. One auxiliary function is defined.

```
/** convenience function to help creating OntologyClass objects. */
 public static OntologyClass ontologyClass(String id, String label) {
     return OntologyClass.newBuilder()
             .setId(id)
             .setLabel(label)
             .build();
 }
```

With this, presented below is a function that creates a Phenopacket that represents the case report described above:

```
public Phenopacket spherocytosisExample() {
      final String PROBAND_ID = "PROBAND#1";
      PhenotypicFeature spherocytosis = PhenotypicFeature.newBuilder()
              .setType(ontologyClass("HP:0004444", "Spherocytosis"))
              .setClassOfOnset(ontologyClass("HP:0011463", "Childhood onset"))
              .build();
      PhenotypicFeature jaundice = PhenotypicFeature.newBuilder()
              .setType(ontologyClass("HP:0000952", "Jaundice"))
              .setClassOfOnset(ontologyClass("HP:0011463", "Childhood onset"))
              .build();
      PhenotypicFeature splenomegaly = PhenotypicFeature.newBuilder()
              .setType(ontologyClass("HP:0001744", "Splenomegaly"))
              .setClassOfOnset(ontologyClass("HP:0011463", "Childhood onset"))
              .build();
      PhenotypicFeature notHepatomegaly = PhenotypicFeature.newBuilder()
              .setType(ontologyClass("HP:0002240", "Hepatomegaly"))
              .setNegated(true)
              .build();
      PhenotypicFeature reticulocytosis = PhenotypicFeature.newBuilder()
              .setType(ontologyClass("HP:0001923", "Reticulocytosis"))
              .build();

      VcfAllele vcfAllele = VcfAllele.newBuilder()
              .setGenomeAssembly("GRCh37")
              .setChr("8")
              .setPos(41519441)
              .setRef("G")
              .setAlt("A")
              .build();

      Variant ANK1_variant = Variant.newBuilder()
              .setVcfAllele(vcfAllele)
              .setZygosity(ontologyClass("GENO:0000135", "heterozygous"))
              .build();

      Individual proband = Individual.newBuilder()
              .setSex(Sex.FEMALE)
              .setId(PROBAND_ID)
              .setAgeAtCollection(Age.newBuilder().setAge("P27Y3M").build())
              .build();

      MetaData metaData = MetaData.newBuilder()
              .addResources(Resource.newBuilder()
                      .setId("hp")
                      .setName("human phenotype ontology")
                      .setNamespacePrefix("HP")
                      .setIriPrefix("http://purl.obolibrary.org/obo/HP_")
                      .setUrl("http://purl.obolibrary.org/obo/hp.owl")
                      .setVersion("2018-03-08")
                      .build())
              .addResources(Resource.newBuilder()
                      .setId("geno")
                      .setName("Genotype Ontology")
                      .setNamespacePrefix("GENO")
                      .setIriPrefix("http://purl.obolibrary.org/obo/GENO_")
                      .setUrl("http://purl.obolibrary.org/obo/geno.owl")
                      .setVersion("19-03-2018")
                      .build())
              .setCreatedBy("Example clinician")
              .build();

      return Phenopacket.newBuilder()
              .setSubject(proband)
              .addPhenotypicFeatures(spherocytosis)
              .addPhenotypicFeatures(jaundice)
              .addPhenotypicFeatures(splenomegaly)
              .addPhenotypicFeatures(notHepatomegaly)
              .addPhenotypicFeatures(reticulocytosis)
              .addAllVariants(ImmutableList.of(ANK1_variant))
              .setMetaData(metaData)
              .build();
  }
```

Messages can be written in binary format using the native protobuf encoding. While this is useful for machine-to-machine communication due to low latency and overhead of serialization, it is not human-readable. An example of writing to an OutputStream is shown here:

```
Path path = Paths.get("/path/to/file");
try (OutputStream outputStream = Files.newOutputStream(path)) {
    Phenopacket phenoPacket = new PhenoPacketExample().spherocytosisExample();
    phenoPacket.writeTo(outputStream);
} catch (IOException e) {
    e.printStackTrace();
}

// read it back again
try (InputStream inputStream = Files.newInputStream(path)) {
    Phenopacket deserialised = Phenopacket.parseFrom(inputStream);
} catch (IOException e) {
    e.printStackTrace();
}
```

In many situations it may be desirable to export the Phenopacket as JSON. This is easy with the following commands:

First add the protobuf-java-util dependency to your Maven POM.xml

```
<dependency>
    <groupId>com.google.protobuf</groupId>
    <artifactId>protobuf-java-util</artifactId>
    <version>${protobuf.version}</version>
    <scope>test</scope>
</dependency>
```

Then use it to print out JSON using the JsonFormat class.

```
Phenopacket p = spherocytosisExample();
try {
    String jsonString = JsonFormat.printer().includingDefaultValueFields().print(p);
    System.out.println(jsonString);
} catch (Exception e) {
    e.printStackTrace();
}
```

A.2.2 Setting up the Java Build

Most users of phenopackets-schema in Java should use maven central to include the phenopackets-schema package. To include the phenopackets-schema package from maven central, add the following to the pom file:

Define the phenopackets.version in the properties section of the pom.xml file.

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  ...
  <phenopackets.version>1.0.0</phenopackets.version>
</properties>
```

Put the following lines of code into the dependencies section of the maven pom.xml file:

```
<dependency>
  <groupId>org.phenopackets</groupId>
  <artifactId>phenopacket-schema</artifactId>
  <version>${phenopackets.version}</version>
</dependency>
```

Users can also download phenopackets-schema from its GitHub repository and install it locally:

```
$ git clone https://github.com/phenopackets/phenopacket-schema
$ cd phenopacket-schema
$ mvn compile
$ mvn install
```

A.2.3 Exporting and Importing Phenopackets

Phenopackets can be easily exported in JSON, YAML and protbuf format. The data should be stored in a datastore with a schema relevant to requirements and be able to map that to the relevant Phenopacket message types for exchange with users/partners. If not stored in a datastore, it is possible that breaking changes to the schema will mean that data cannot be exchanged with parties using a later version of the schema or if the schema is updated, tools will no longer be able to read data written using the previous version. While protobuf allows for 'schema evolution' by design, which will limit the impact of changes to the schema precipitating this scenario, it is nonetheless a possibility.

JSON Export

In many situations it may be desirable to export the Phenopacket as JSON. This can be done with the following commands:

```
import org.phenopackets.schema.v1.Phenopacket;
import com.google.protobuf.util.JsonFormat;
import java.io.IOException;
```

```
Phenopacket phenoPacket = // create a Phenopacket
try {
    String jsonString = JsonFormat.printer().includingDefaultValueFields().print(pp);
    System.out.println(jsonString);
 } catch (IOException e) {
   e.printStackTrace();
 }
```

YAML Export

YAML (YAML Ain't Markup Language) is a human friendly data serialization standard for all programming languages.

```
import org.phenopackets.schema.v1.Phenopacket;
import com.google.protobuf.util.JsonFormat;
import java.io.IOException;
import com.fasterxml.jackson.dataformat.yaml.YAMLMapper;

Phenopacket phenoPacket = // create a Phenopacket
try {
    String jsonString = JsonFormat.printer().includingDefaultValueFields().
print(phenoPacket);
    JsonNode jsonNodeTree = new ObjectMapper().readTree(jsonString);
    String yamlString = new YAMLMapper().writeValueAsString(jsonNodeTree);
    System.out.println(yamlString);
} catch (IOException e) {
    e.printStackTrace(); // or handle the Exception as appropriate
}
```

Protobuf Export

For most use case, it is recommended to use JSON as the serialization format for Phenopackets. Protobuf is more space efficient than JSON, but it is a binary format that is not human readable. It is possible to write to any OutputStream (replace System.out in the below code).

```
import org.phenopackets.schema.v1.Phenopacket;
import java.io.IOException;

Phenopacket phenoPacket = // create a Phenopacket
try {
    phenoPacket.writeTo(System.out);
 } catch (IOException e) {
    e.printStackTrace(); // or handle the Exception as appropriate
}
```

Importing Phenopackets (JSON format)

There are multiple ways of doing this with different JSON libraries, e.g. Jackson, Gson, JSON.simple. The following code explains how to convert the JSON String object into a protobuf class. This is not limited to a Phenopacket message, so long as the type of message contained in the json is known, it can be merged into the correct Java representation.

```
String phenopacketJsonString = // Phenopacket in JSON as a String;
try {
    Phenopacket.Builder phenoPacketBuilder = Phenopacket.newBuilder();
    JsonFormat.parser().merge(jsonString, phenoPacketBuilder);
    Phenopacket phenopacket = phenoPacketBuilder.build();
    // do something with phenopacket ...
} catch (IOException e1) {
    e1.printStackTrace(); // or handle the Exception as appropriate
}
```

A.2.4 Evidence

The evidence code is used to document the support for an assertion. Below is an example for the assertion that flexion contractures are found in stiff skin syndrome.

```
import org.phenopackets.schema.v1.core.Evidence;
import org.phenopackets.schema.v1.core.ExternalReference;
import org.phenopackets.schema.v1.core.OntologyClass;

OntologyClass publishedClinicalStudy = OntologyClass.
            newBuilder().
            setId("ECO:0006017").
            setLabel("author statement from published clinical study used in manual
assertion").
            build();
    ExternalReference reference = ExternalReference.newBuilder().
            setId("PMID:20375004").
            setDescription("Mutations in fibrillin-1 cause congenital scleroderma: stiff
skin syndrome").
            build();
    Evidence evidence = Evidence.newBuilder().
            setEvidenceCode(publishedClinicalStudy).
            setReference(reference).
            build();
```

This code produces the following Evidence element:

```
{
    evidence_code {
        id: "ECO:0006017"
        label: "author statement from published clinical study used in manual assertion"
    }
    reference {
        id: "PMID:20375004"
        description: "Mutations in fibrillin-1 cause congenital scleroderma: stiff skin
syndrome"
    }
}
```

A.2.5 Timestamp

A Timestamp represents a point in time independent of any time zone or local calendar, encoded as a count of seconds and fractions of seconds at nanosecond resolution. The count is relative to an epoch at UTC midnight on January 1, 1970, in the proleptic Gregorian calendar that extends the Gregorian calendar backwards to year one.

A timestamp is required for several elements of the Phenopacket, including the MetaData. Usually, code will create a timestamp to represent the current time (the time at which the Phenopacket is being created).

```
import com.google.protobuf.Timestamp;

long millis  = System.currentTimeMillis();
Timestamp timestamp = Timestamp.newBuilder().setSeconds(millis / 1000)
            .setNanos((int) ((millis % 1000) * 1000000)).build();
```

It is also possible to create a timestamp for an arbitrary date, see example below:

```
import com.google.protobuf.Timestamp;
import java.text.SimpleDateFormat;
import java.util.Date;

SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
String hastings = "1066-10-14";
Date date = formatter.parse(hastings);
long millis = date.getTime();
Timestamp timestamp = Timestamp.newBuilder().setSeconds(millis / 1000)
          .setNanos((int) ((millis % 1000) * 1000000)).build();
```

If more precision is desired, use the following format:

```
SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
```

A.2.6 Duration

The Age messages use ISO 8601 duration strings. These can be easily converted to Java types using the Period class.

```
import java.time.Period;

Subject subject = phenopacket.getSubject();
if (subject.hasAgeAtCollection()) {
    // Phenopacket Age
    Age ageAtCollection = subject.getAgeAtCollection();
    // Java Period
    Period agePeriod = Period.parse(ageAtCollection.getAge());
}
```

A.3 Working with Phenopackets in C++

A.3.1 Generating C++ Files

The maven build generates Java, C++, and Python code that can be directly used in other projects. Therefore, if you have maven set up on your machine, the easiest way to generate the C++ files is:

```
$ mvn compile
$ mvn package
```

This will generate four files in the following location:

```
$ ls target/generated-sources/protobuf/cpp/
    base.pb.cc          phenopackets.pb.cc
    base.pb.h           phenopackets.pb.h
```

The other option is to use Google's protoc tool to generate the C++ files. The following commands will generate identical files in a new directory called gen:

```
$ mkdir gen
$ protoc \
    --proto_path=src/main/proto/ \
    --cpp_out=gen/ \
    src/main/proto/phenopackets.proto src/main/proto/base.proto
```

The protoc command specifies the directory where the protobuf files are located (–proto_path), the location of the directory to which the corresponding C++ files will be written, and then passes the two protobuf files.

A.3.2 Compiling and Building Phenopackets

The phenopacket code can be compiled and built using standard tools. Here is a small example of a C++ program that reads in a phenopacket JSON file from the command line and prints our some of the information contained in it to the shell. The classes defined by the phenopacket are located within namespace declarations that mirror the Java package names, and thus are extremely unlikely to collide with other C++ identifiers.

```
#include <iostream>
#include <string>
#include <fstream>
#include <sstream>

#include <google/protobuf/message.h>
#include <google/protobuf/util/json_util.h>

#include "phenopackets.pb.h"


using namespace std;
```