Freie Universität Berlin
Fachbereich Mathematik und Informatik

Implementation of Phenopacket import and export functions for patient phenotyping in the Symptom Annotation Made Simple (SAMS) application

Bachelorarbeit Bioinformatik

Florian Herzler
Matrikel-Nr. 5214040
florian.herzler@gmail.com

Berlin, 9. Mai 2022

Gutachter: Prof. Dr. Dominik Seelow
Zweitgutachter: Prof. Dr. Sigmar Stricker
Betreuer: Robin Steinhaus

SELBSTSTÄNDIGKEITSERKLÄRUNG

Name:	Herzler
Vorname:	Florian
Studiengang:	Bioinformatik BSc
Matrikelnummer:	5214040

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Bachelorarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.
Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.
Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keiner anderen Universität als Prüfungsleistung eingereicht.

Datum: 09.05.2022 Unterschrift: F. Herzler

Zusammenfassung

Viele Menschen, die an monogenen Erkrankungen leiden, müssen teilweise lange auf eine korrekte medizinische Diagnose warten. Eine genaue Dokumentation des vorhandenen Phänotyps dieser Patienten kann die Präzision der medizinischen Behandlung verbessern.
SAMS (Symptom Annotation Made Simple) zielt darauf ab, diese genaue Phänotypisierung mit einer intuitiven webbasierten Anwendung in die klinische Routineversorgung einzubringen, wobei Standard-Annotationssysteme wie die Human Phenotype Ontology (HPO), Online Inheritance In Man (OMIM) und Orphanet verwendet werden. SAMS speichert die genauen Phänotypdaten von Patienten und ist ein Hilfsmittel, das Ärzte bei der Diagnose und der besseren Behandlung von Patienten unterstützt, wobei das Hauptaugenmerk auf seltenen Krankheiten liegt.
Diese Arbeit implementiert Funktionen für den Im- und Export von *Global Alliance for Genomics and Health* Phenopackets um die Kommunikation zwischen und die Zusammenarbeit von verschiedenen medizinischen und Forschungsabteilungen zu verbessern.

Abstract

Many people suffering from single-gene disorders have to wait a significant amount of time for a correct medical diagnosis. An accurate documentation of the present phenotype of these patients can improve the precision of medical treatment.

SAMS (Symptom Annotation Made Simple) aims to bring this phenotyping to routine clinical care with an intuitive web-based application, using standard annotation systems such as the Human Phenotype Ontology (HPO), Online Inheritance In Man (OMIM) and Orphanet. Storing patients' precise phenotype data, SAMS is a tool to assist physicians with reaching a diagnosis and providing better treatment to patients, focusing primarily on rare diseases.

This work implements functions for the import and export of SAMS' patient data as *Global Alliance for Genomics and Health* Phenopackets to improve the communication between and cooperation of different medical and research departments.

Contents

Todo list

1 Introduction

The European Commission defines a disease as rare if it affects no more than 0.05% of the population (1 in 2,000), totaling the number of affected persons in the EU at approximately 30 million[1].

"[A]lthough rare diseases are individually rare by definition, they are collectively common" [Ferreira, 2019]. Individuals suffering from rare diseases (RDs) often unwillingly embark on a so-called diagnostic odyssey[2] until they are diagnosed correctly [EURORDIS, 2007; Cherif et al., 2014]. This may include unnecessary medical procedures and even wrong diagnoses while often lasting multiple years[3]. In addition to emotional stress it also delays the timely start of appropriate medical treatment.

The aim of precision medicine is the stratification into patient disease subgroups based on diseases in order to improve diagnosis and subsequent medical treatment, as stated by [M. A. Haendel et al., 2018]. This is not only useful for rare disease communities but also for e.g. cancer therapy [Jacobsen et al., 2021].

A substantial part in the process of precision medicine (PM) is the concept of deep phenotyping [König et al., 2017], which can be defined as the *"precise and comprehensive analysis of phenotypic abnormalities in which the individual components of the phenotype are observed and described"* [Robinson, 2012].

Robinson defines a phenotype as a collection of phenotypic signs (e.g. clinical abnormalities but also hair color) that often refer to a deviation from the norm.

Figure 1.1 visualizes the connection between these medical concepts. Solid lines signify the smaller circle being a sub concept of the encompassing circle while the dashed line signifies the use of deep phenotyping (DP) to identify diseases.

[1] https://ec.europa.eu/info/research-and-innovation/research-area/health-research-and-innovation/rare-diseases_en (accessed 2022-05-07)

[2] https://www.globalrarediseasecommission.com/AboutUs (accessed 2022-05-08)

[3] https://ncats.nih.gov/programs/diagnostic-odyssey (accessed 2022-05-08)

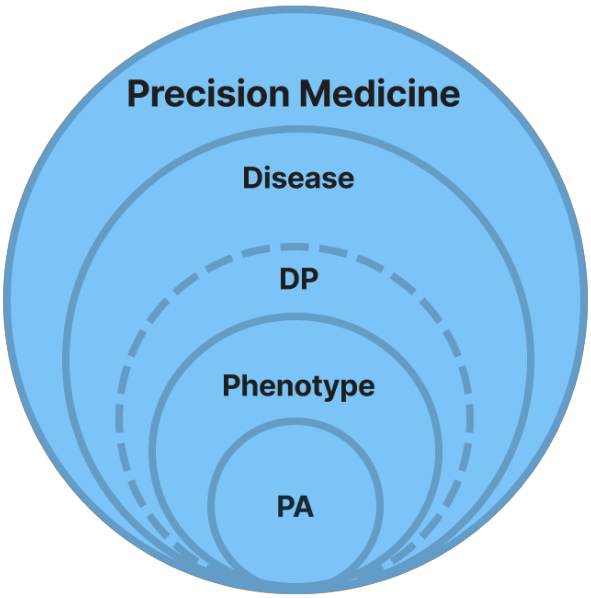


Figure 1.1: Precision medicine involves assigning patients to subgroups based on diseases. Diseases are diagnosed using deep phenotyping (DP). DP identifies phenotypes consisting of phenotypic abnormalities (PA).

A medical diagnosis attempts to classify the symptoms and signs that a patient presents into categories (diseases)[4]. Patients presenting different clinical signs while being diagnosed with the same diagnosis can therefore exist just as patients can receive different diagnoses while presenting the same clinical signs.

The ICD (International Classification of Disease)[5] codes are the basis for comparable statistics on causes of non-fatal diseases and death and are adopted globally by medical facilities. They can be described as the infrastructure for clinical and health information [Harrison et al., 2021] and are referred to as the basis of so called *billing diagnoses*, as used in [Steinhaus et al., 2022].

In 2007, EURORDIS conducted a survey including 8 RDs (such as Marfan syndrome and Crohn's disease), discovering that 40 % of patients initially received a false diagnosis while 25% waited between 5 and 30 years for a correct diagnosis while communication with patients or, if a child is affected, their parents was insufficient and made coping with the disease more difficult [EURORDIS, 2007; Rosenthal et al., 2001].

Attainment speed of diagnosis varies between RDs, with some being easier and faster to diagnose. Cystic fibrosis for example can be diagnosed quite easily with a sweat test that checks

[4] https://en.wikipedia.org/wiki/Medical_diagnosis (accessed 2022-05-08)
[5] https://www.who.int/standards/classifications/classification-of-diseases (accessed 2022-05-08)

chloride levels and has recently been implemented as a newborn screening in the United States[6].

More accurate capture of presented phenotypic abnormalities can thus contribute to the overcoming of two medical problems:

First, it can assist physicians with differential diagnosis for rare diseases (RD) and improve the speed thereof [Robinson, 2012].
This is especially difficult with the estimated amount of known RDs being over 10,000 (a more conservative amount is approx. 7,000[6]) [M. Haendel et al., 2020].

Second, it can be helpful even after having identified the correct diagnose. A problem with this ICD *billing diagnosis* commonly used by clinical information systems is the loss of information about the clinical signs an individual actually presented. Symptoms can change over time, but an underlying genetic disorder will not vanish. Knowledge of the correct diagnosis is imperative for a targeted medical plan, but this alone is not sufficient. The phenotypic abnormalities have to be documented thoroughly to decide about medication or procedures and avoid unnecessary medical actions.

1.0.1 Marfan syndrome

The Marfan syndrome (MFS) is a rare autosomal dominant disorder, affecting approximately 1 in 10,000 individuals whith no *"geographic, ethnic or gender predilection"*[Kumar and Agarwal, 2014; Yuan and Jing, 2010].
It is a systemic genetic disease affecting the connective tissue, predominantly the cardiovascular, ocular and musculosceletal system, but also the lungs [Radonic et al., 2010]. *"The majority of cases of MFS (MFS1) are caused by a mutation in the fibrillin-1 gene (FBN1) on chromosome 15 (15q21.1)"*, affecting the elasticity of connective tissue [Kumar and Agarwal, 2014].
"A syndrome is a set of medical signs and symptoms which are correlated with each other and often associated with a particular disease or disorder"[7].
MFS occurs with a high degree of clinical variability and the diagnosis is based on the *Ghent nosology*, defining a set of clinical criteria [B. Loeys et al., 2001] that confirms the diagnosis in over 95% of patients [B. L. Loeys et al., 2010]. The primary criteria are *aortic root aneurysm*[8] and *ectopia lentis*[9].

[6] https://www.nhlbi.nih.gov/health/cystic-fibrosis/diagnosis (accessed 2022-05-08)
[7] https://en.wikipedia.org/wiki/Syndrome (accessed 2022-05-07)
[8] a bulge in the aorta, possibly resulting in a rupture
[9] dislocation of the lens

MFS was chosen as an example for this thesis because of its phenotypic variety and the presence of related conditions resulting from gene FBN-1 (*Weill-Marchesani syndrome*) or present similar phenotypes (*Loeys-Dietz syndrome*, gene TGFBR1/2). Figure 5 has been adopted from [B. L. Loeys et al., 2010] and can be found in the Appendix.

1.1 Symptom Annotation Made Simple - SAMS

Symptom Annotation Made Simple (SAMS), developed at Berlin Institute of Health at Charité (BIH) group for Bioinformatics and Translational Genetics, is a web-based phenotyping tool aimed at symptom-based deep phenotyping in medical care and research in order to improve the accuracy and speed of diagnoses as well as the pathophysiological understanding of underlying diseases. It targets RD communities but is not limited to only those use cases [Steinhaus et al., 2022].
SAMS offers a selection of functions for phenotyping patients using diagnoses or diseases from databases Orphanet [Pavan et al., 2017] and Online Mendelian Inheritance in Man (OMIM) [Amberger et al., 2019], German Alpha-ID diagnose codes as well as clinical signs or symptoms from the Human Phenotype Ontology (HPO) [Groza et al., 2015].
Figure 1.3 shows a visualization of SAMS and its associated data sources, with Phenopackets being covered in Chapter 1.3.

1.2 Data sources of SAMS

1.2.1 OMIM

Online Mendelian Inheritance in Man (OMIM) is the digital continuation of *Mendelian Inheritance in Man* (MIM), a series of 12 publications between 1966 and 1998, serving as a collection of knowledge about (mono)genetic disorders and associated phenotypes.
Entries are assigned a unique identifying `MIM` number (e.g.Loeys-Dietz syndrome is assigned 6092192 (OMIM:609192)), with the source of information being peer-reviewed biomedical literature [Amberger et al., 2019].
OMIM entries are linked to a *Clinical Synopsis*, a list of associated terms from external data sources, including the HPO, ICD-10 terms and Orphanet. Hidden per default, these external links can be displayed by checking the `Feature IDs` button at the top of the page next to the search bar. An example for the *Clinical Synopsis* of *Marfan syndrome* can be accessed at https://omim.org/clinicalSynopsis/154700 or seen in Figure 5.4.

OMIM includes [Amberger et al., 2019]:

- single gene Mendelian disorders (e.g., Marfan syndrome, achondroplasia, Huntington disease)
- traits (e.g. hair and eye color)
- susceptibility to drug reaction (e.g., malignant hyperthermia, warfarin sensitivity)
- altered susceptibility or reaction to infection (herpes simplex encephalitis, progression of HIV infection to AIDS)
- germline susceptibility to cancer (e.g. BRCA1 and breast/ovarian cancer)

1.2.2 Alpha-ID

The German Institute of Medical Documentation and Information (*Deutsches Institut für Medizinische Dokumentation und Information - DIMDI*) introduced the diagnose code Alpha-ID as a prototype in 2005 based on ICD-10-GM (german modification of ICD version 10), refining the broader ICD codes for German healthcare[10]. They are actively maintained by DIMDI's legal successor Federal Institute for Drugs and Medical Devices (BfArM), which releases valid versions for each calendar year [11].

Alpha-ID-Code	ICD-10-GM-Code	Entry
I6158	L30.8	Ichthyosiformes Ekzem
I6159	L30.8	Nässendes Ekzem
I6154	L30.8	Nässender Nabel
I6142	L30.8	Trockenes Ekzem

Table 1.1: Refined Alpha-IDs for the same ICD-10 code obtained from BfArM

SAMS uses the extension **Alpha-ID-SE** for rare diseases (German: "Seltene Erkrankungen"), which add the Orphanet code of the disease to the Alpha-ID, in essence adding a third ID column to the layout of Table 1.1. SAMS uses `Alpha-ID-SE` codes, so any Alpha-IDs mentioned in this thesis are referring to the rare disease extension. As they are specific to German healthcare and are not in SAMS' focus at the moment they will not be covered in the main part of this thesis.

[10] https://www.bfarm.de/EN/Code-systems/Terminologies/Alpha-ID-SE/_node.html (accessed 2022-05-06)
[11] Provided at BfArM

1.2.3 HPO

The Human Phenotype Ontology (HPO) was introduced in 2008 *"to provide a comprehensive logical standard to describe and computationally analyze phenotypic abnormalities found in human disease"* [Robinson et al., 2008].
As an ontology, it structures medical phenotypes into subcategory levels of increasing depth, enabling precise definitions of phenotypes. It is a Web Ontology Language (OWL)[12], making sophisticated computational analysis of its over 15,000 terms possible because of logical inference [Köhler et al., 2021; M. A. Haendel et al., 2018].
The HPO achieves this by being structured as a Directed Acyclic Graph (DAG)[13] with directed edges and nodes that cannot form cycles in the resulting graph. Thus, a child node can have

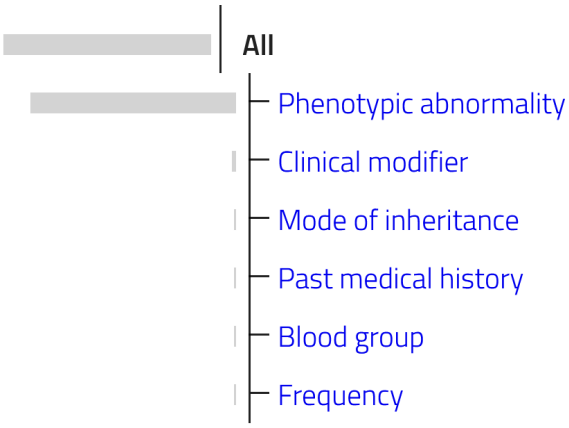


Figure 1.2: Screenshot of the HPO's top-level subontologies. The gray bar represents the proportion of contained terms with the most being *Phenotypic abnormalities*.

multiple parents but no child can be its own parent, resulting in a leveled structure of terms categorized into increasingly specific *child* **is-a** *parent* relations (e.g. *Abnormality of the musculature* **is-a** *Abnormality of the musculoskeletal system* **is-a** *Phenotypic abnormality* in Figure 5.3).
The HPO's top level subontologies act as the primary distinguishing categories and can be seen in Figure 1.2.
Each term possesses a unique ID (e.g. HP:0003011 for *Abnormality of the musculature*) and a list of assigned *Disease Associations*, which contains links to OMIM or Orphanet entries.
By clicking on one of these IDs or searching the disease in the search bar, a new list of (clinical) signs (HPO annotations, HPOA) appears. If specific studies are linked, frequencies of the phenotype are displayed in either a numerical format (*69/420*) or textual context (one of *Very*

[12] W3C Web Ontology Language
[13] Wikipedia entry

rare - Occasional - Frequent - Very frequent). HPO annotations for OMIM were created using text mining of the *Clinical Synopsis* section mentioned in 1.2.1 and curation of the HPO team, which led to a total of 108 580 annotations with a mean of 13.9 HPO annotations per OMIM disease [Köhler et al., 2021].
An example can be seen in Table 5.2.
Unfortunately, the CSV export function offered by the HPO only exports the *HPO_TERM_ID*, *HPO_TERM_NAME* and the *CATEGORY* (e.g. *Skeletal System*), but not the observed frequencies, so one would have to process the provided text file. Table 5.1 shows the first half of the exported CSV data for *Marfan syndrome*[14].
The HPO also offers `.hpoa` annotation files on their website[15] and in addition already linked `.txt` files (`genes_to_phenotype.txt` and `phenotype_to_genes.txt`, stating the ORPHA or OMIM ID alongside frequency and gene data).
Orphanet has also added annotations using the HPO to annotate RDs, as mentioned in Chapter 1.2.4. Köhler et al. reported the usage of 7495 HPO terms to annotate 3956 rare diseases, resulting in an average of 24.4 terms per disease (96 612 annotation in total at the time of their publication in 2021). I could not find an Orphanet publication that explains this process, so this information was derived from Köhler et al.'s publication about the HPO in 2021.

1.2.4 Orphanet

The Orphanet Database (Orphanet) is an international database for rare disease (RD) resources which, in contrast to OMIM, do not have to be of genetic origin. Orphanet was created in 1997 by the French National Institute of Health and Medical Research (Institut national de la santé et de la recherche médicale - Inserm) as a tool to bundle fragmented data about RDs and their treatment with *Orphan Drugs*. These are drugs used *"to treat diseases so rare that sponsors are reluctant to develop them under usual marketing conditions"* [16].
Orphanet also classifies terms with a unique ID (ORPHA:558 for Marfan syndrome) and displays additional information (external OMIM and ICD-10 codes, inheritance, onset and classification level) below a short description of the disease (Figure 5.5).
Orphanet has later introduced annotations using the HPO to annotate rare diseases with frequencies of phenotypic abnormalities.

[14] full overview at https://hpo.jax.org/app/browse/disease/OMIM:154700
[15] HPO annotation download section
[16] https://www.orpha.net/consor/cgi-bin/Education_AboutOrphanDrugs.php?lng=EN

The frequency in the patients' population can be :

- always present: 100%
- very frequent: 99% - 80%
- frequent: 79% - 30%
- occasional: 29% - 5%
- rare: 4% - 1%

The phenotypic abnormality can be defined as one of the following :

- Pathognomonic sign : a sign whose presence indicates that a particular disease is present beyond any doubt. The absence of this sign does not exclude the possibility of the presence of the disease, but the presence of the pathognomonic sign affirms it with certainty.
- Diagnostic criterion : phenotypic abnormalities noted as « diagnostic criterion » are those included in established sets of criteria to establish the diagnosis of a particular disease having been published in a peer-reviewed journal.
- Exclusion criterion : phenotypic abnormalities noted as « exclusion criterion » are those that are always absent in a particular disease and therefore exclude its diagnosis.

The last paragraph has been adopted identically from the Orphanet website.

1.3 Phenopacket Schema

The Phenopacket Schema, an open standard approved by the Global Alliance For Genomics and Health (GA4GH) in 2019[17], was created to improve the communication between research and clinical communities by linking disease and phenotype information to patient and genetic information. It is a set of rules, specifying the organization of data, thus creating a template for a Phenopacket:
"[A] structured representation of an individual's medically relevant data, providing a computable case report of either a single medical encounter or a time course that can represent the entire medical history of an individual" [Jacobsen et al., 2021].
In contrast to genomic research, no standard for exchanging phenotypic data was widely adopted

[17] Link to announcement

previously, searching for phenotypic data led to many storage sites ranging from scientific publications over Electronic Health Records (EHR) to patient health forums [Jacobsen et al., 2021]. Phenopackets aim at providing that interchangeable format used interdisciplinary to ensure optimal cooperation of medical and research fields.

This schema allows common medical ontologies like the HPO, Mondo Disease Ontology

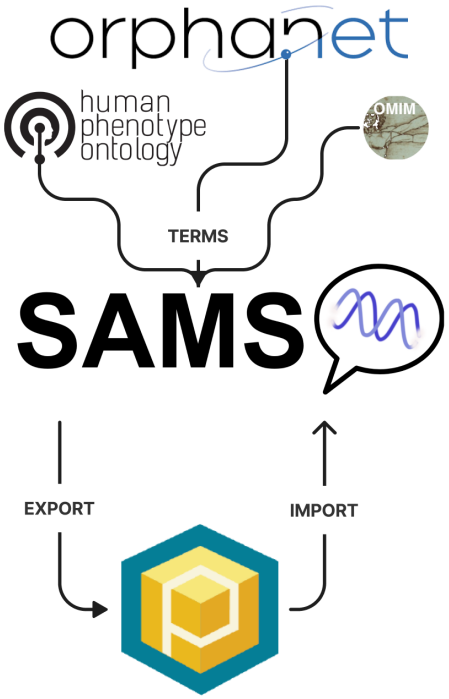


Figure 1.3: Data flow between data sources, SAMS and Phenopackets.

(MONDO)[18] and Orphanet Rare Disease Ontology (ORDO) to cover a wide base of clinical phenotype use cases and provide a framework in which ontologies and database data can be exchanged between different systems and medical disciplines.

Because of the appearance in either JSON[19] or YAML[20] files, Phenopackets bridge the gap between human and machine readability, even enabling inexperienced users to quickly understand the underlying principles of the different aspects in a Phenopacket.

Encouraging DP in addition to broader information [21] and incorporating external resources like genomic information (e.g. Variant Call Format (VCF) files) makes Phenopackets a flexible standard, that is easily configured to different medical applications.

[18] https://mondo.monarchinitiative.org (accessed 2022-05-08)
[19] JavaScript Object Notation (JSON)
[20] YAML Ain't Markup Language (YAML)
[21] e.g. *Abnormal hair morphology* (HP:0001595) or *Partial duplication of eyebrows* (HP:0045018)

1.3.1 Multiplicity and Requirement Levels

Before elaborating on any of the elements, I will quickly define the terminology used to describe the requirement levels. This section is largely adopted from the Phenopackets documentation for requirement levels[22].
Requirement levels are `OPTIONAL, RECOMMENDED and REQUIRED` and only apply if the element in question is being used.
If, as an example, `exampleElement` is not present, the requirements for child element `exampleChild` do not take hold.
These requirement levels have to be validated in the respective application, that way different use cases can be achieved (e.g. requiring a `disease` element when working with rare diseases). A phenopacket validation software has to reject an instance without the necessary `REQUIRED` parameters and might emit a warning message for missing `RECOMMENDED` parameters. `OPTIONAL` elements may be handy in certain situations but clutter the phenopacket in other circumstances and can thus be omitted.

1.3.2 Top-Level Elements

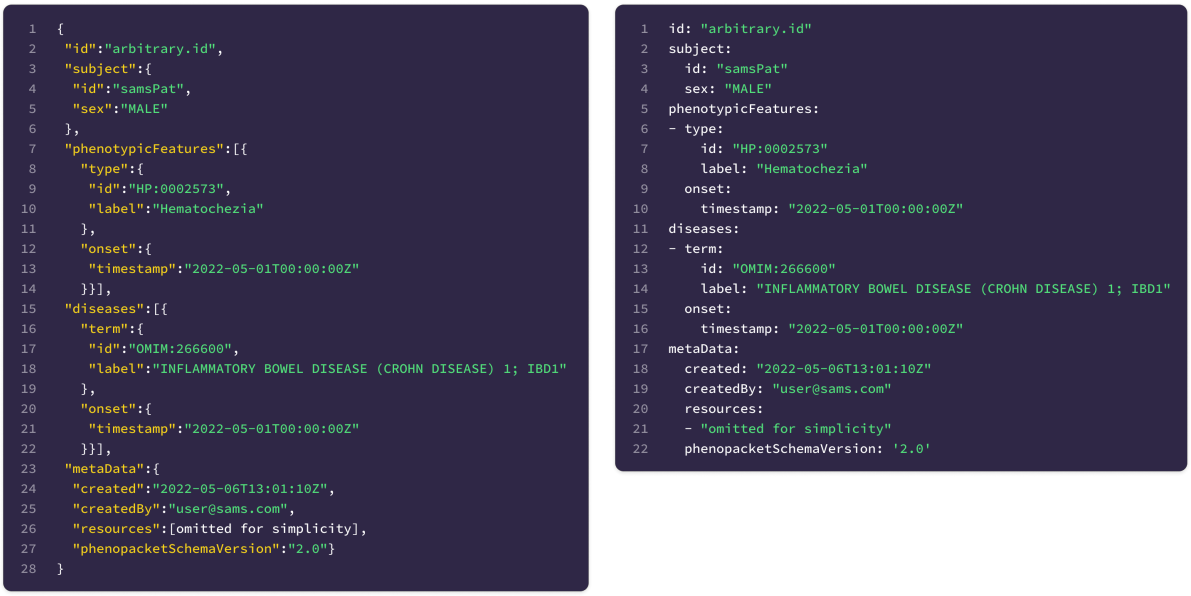
The Phenopacket schema offers three top-level elements. **The Phenopacket** includes the most sub categories and is, as the name implies, the heart of the schema. Its components will be explained below. **The Family element** can connect a `proband` phenopacket with optional phenopackets of `relatives`, a `REQUIRED` `pedigree` and `OPTIONAL` `files` related to the entire family. **Cohorts** describe a collection of `members` phenopackets related in a phenotypic or genotypic aspect, for example as part of a clinical study and provide `OPTIONAL` `files` related to the whole cohort[23].

1.3.3 Phenopacket building blocks

The Phenopacket schema allows many elements in a Phenopacket to be optional. Elements can therefore be viewed as building blocks, which can be added during the construction of a Phenopacket. An overview of all officially supported Phenopacket elements can be found in Figure 5.6 from Köhler et al., more extensive versions can be found in their documentation as detailed and overview diagrams. In Chapter 2.4 I will highlight the building blocks used in SAMS as part of this thesis.

[22] https://phenopacket-schema.readthedocs.io/en/latest/requirements.html (accessed 2022-05-06)
[23] https://phenopacket-schema.readthedocs.io/en/latest/toplevel.html (accessed 2022-05-04)

Most of the components are optional, so a valid phenopacket can be relatively small (Figure 1.4).



```json
{
 "id":"arbitrary.id",
 "subject":{
  "id":"samsPat",
  "sex":"MALE"
 },
 "phenotypicFeatures":[{
   "type":{
    "id":"HP:0002573",
    "label":"Hematochezia"
   },
   "onset":{
    "timestamp":"2022-05-01T00:00:00Z"
   }}],
 "diseases":[{
   "term":{
    "id":"OMIM:266600",
    "label":"INFLAMMATORY BOWEL DISEASE (CROHN DISEASE) 1; IBD1"
   },
   "onset":{
    "timestamp":"2022-05-01T00:00:00Z"
   }}],
 "metaData":{
  "created":"2022-05-06T13:01:10Z",
  "createdBy":"user@sams.com",
  "resources":[omitted for simplicity],
  "phenopacketSchemaVersion":"2.0"}
}
```

```yaml
id: "arbitrary.id"
subject:
  id: "samsPat"
  sex: "MALE"
phenotypicFeatures:
- type:
    id: "HP:0002573"
    label: "Hematochezia"
  onset:
    timestamp: "2022-05-01T00:00:00Z"
diseases:
- term:
    id: "OMIM:266600"
    label: "INFLAMMATORY BOWEL DISEASE (CROHN DISEASE) 1; IBD1"
  onset:
    timestamp: "2022-05-01T00:00:00Z"
metaData:
  created: "2022-05-06T13:01:10Z"
  createdBy: "user@sams.com"
  resources:
  - "omitted for simplicity"
  phenopacketSchemaVersion: '2.0'
```

Figure 1.4: Simple example showing the differences between JSON and YAML for the same data. The male patient *samsPat* presented *Crohn's disease* and *Hematochezia* on 2022-05-01. For simplicity I removed the `resources`.

2 Background

2.1 Programs functionally related to Symptom Annotation Made Simple

2.1.1 PhenoTips

PhenoTips is a web-based application, available at https://phenotips.com. It started out as an open-source project but is now focused on a closed-source enterprise version.
PhenoTips enables users to document *"clinical phenotype, genetic, disease, and family history data for patients with genetic diseases"*, offering terms from the HPO for phenotypic abnormalities as well as OMIM and Orphanet for diagnoses[1] [Girdea et al., 2013].

2.1.2 Phenotate

Used in medical teaching, the web application *Phenotate* lets its users annotate OMIM and Orphanet disorders with terms from the HPO. With this crowd-sourcing approach, it aims to collect *"phenotype information using student assignments"* at https://phenotate.org [Chang et al., 2020]. A sample annotation provided by *Phenotate* can be accessed through their website[2].

2.2 Functions of Symptom Annotation Made Simple

The following functions explained in the background of this thesis were already present and have not been created by me. My work is documented in the *Methods and Results* in Chapters 3.1.2 and 3.1.3.
SAMS can be used in four different ways:

1. manage patients' phenotypes utilizing the associated database (physicians)

2. self-phenotyping (patients or their parents)

[1] https://github.com/phenotips/phenotips (accessed 2022-05-07)
[2] https://app.phenotate.org/dashboard/in-progress (accessed 2022-05-08)

3. integration into other applications

4. importing and exporting GA4GH Phenopackets (Chapters 3.1.2 and 3.1.3)

Registering as a physician gives the user an overview of their patients as the centerpiece of SAMS (Figure 2.1).

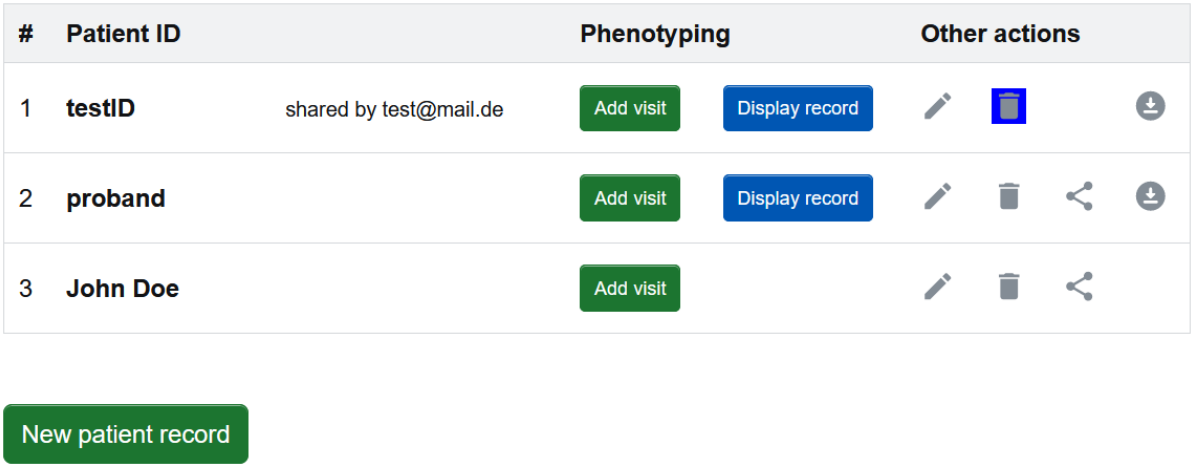


Figure 2.1: SAMS' *Patient Management* with a shared patient and own patients with and without visits.

New patients can be added and existing patients can be edited and deleted. SAMS stores its phenotype data as a list of visits for each patient. Each visit emulates a medical encounter with a physician and is comprised of the visit date and clinical signs or diseases. If saved visits for a patient already exist, physicians can view those as an expandable list by selecting `Display Record`. The same phenotype can appear in multiple, even consecutive visits. A more insightful visualization of the symptom history is shown by clicking on `Time course` on the `Previous Visits` page (Figure 3.3).

With `Import a Phenopacket`, users[3] are presented another option of creating a new patient. Before creation of these, medical doctors can choose from an internally created list of visits and their corresponding `phenotypicFeatures` and `diseases` resulting from the Phenopacket.

An existing patient with at least one visit can be exported as a Phenopacket by clicking the

[3] presumably clinicians, also other healthcare professionals (e.g. diagnostics laboratories)

download icon, while a list of all patients associated with a physician can be exported by clicking the `Export all patients` button. These new functions of SAMS are the result of this thesis and will be discussed further in Chapters 3.1.2 and 3.1.3.
SAMS' physician users can share their patients with other physician users. *Physician A* can share a patient with *physician B* by sending the URL presented in a pop-up after clicking on the share icon. These URLs are valid for 24 hours and appear, after opening the URL as logged-in *physician B*, in the patient management list of *physician B*.
An indication of the shared status of the patient is presented by displaying *shared by physicianA@mail.com* and can be observed in Figure 2.1. Shared patients cannot be deleted, edited or re-shared, but new visits can be added.
New visits for a patient can be added by clicking on `Add visit`, which redirects the user to the central interface of SAMS: `Enter phenotypes` and complete diagnoses for patients. Terms can be searched in a search bar, searching begins when at least three letters have been entered.
Results that match the search term are displayed as a list for each of the selected data sources, which can be unselected. Per default all three (HPO, Orphanet and OMIM) are activated, while DIMDI is deactivated.Terms can be either marked as present or specifically marked as absent with a checkbox in the leftmost column.
Orphanet provides the term description and an ID that links to the term on the Orphanet website.
OMIM additionally displays the abbreviation of a disorder, if present.
The search mechanism for the HPO is more sophisticated, because its inherent structure as an ontology allows for a tree-like search. Broader terms can be refined by expanding the subcategories with the arrow symbol, which can then be hidden again with the crossed-out eye symbol.
This allows for precise phenotyping with the HPO's over 15,000 structured terms [Köhler et al., 2021]. A date for the visit has to be chosen before either exporting the data as a Phenopacket or saving it to the database.

Patient accounts are able to edit their own data, view previous visits and the resulting time course. They can also export these as GA4GH Phenopackets or save it to the SAMS database. SAMS offers an *Use SAMS without Login* mode to test out all of physician utilities as a generic test user. As saved data will be accessible for all test users, SAMS displays a warning (Figure 5.1). In addition, SAMS lets users *Phenotype a patient and create a Phenopacket*, internally labeled as the `on-the-fly` mode.
The relations of SAMS' web pages are visualized as a page flow diagram in Figure 2.3.

2.3 Programming languages and connections

SAMS has been in development for multiple years and therefore already had a solid code foundation, utilizing a combination of the programming languages Perl and JavaScript. The User Interfaces (UIs) are constructed in Hypertext Markup Language (HTML) and styled with Cascading Style Sheets (CSS).
SAMS uses the Database Management System PostgreSQL[4] for communication with and manipulation of stored data.
Tobias Schalau visualized the interconnections in his bachelor thesis in 2020, his diagram can be seen in Figure 5.

Since then, SAMS has started using the Perl module `HTML::Template`[5]. Static, unchanging `.tmpl` files can be called from CGI scripts, allowing the passing on of values from Perl to these templates. This separates the data and UI layers.
The tags used in this thesis are listed in the following section:

`<TMPL_VAR>` This tag replaces the variable in the HTML template when it is called with a corresponding value, as seen in Code 1.

```
<TD>
  <TMPL_VAR DATE>
</TD>
```

Listing 1: An HTML table entry with the value of variable `DATE`.

`<TMPL_IF> and <TMPL_ELSE>` If a variable is 'true' for Perl (in SAMS we use `1`, as internally this gets interpreted as 'false'[6]), then we can check this using

```
<TMPL_IF NAME="PARAMETER">
```

and execute a following block of code if the parameter is 'true'. If it is not, a code block for the 'else' case can be provided, but is not required. `<TMPL_UNLESS>` is the opposite of `<TMLP_IF>` and is executed only if the parameter is false

`<TMPL_LOOP>` This accepts a list of parameter assignments (Perl: `array` reference of `hash` references) and uses each list entry for one iteration of the loop. This is useful for passing

[4] https://www.postgresql.org/ (accessed 2022-05-06)
[5] https://metacpan.org/pod/HTML::Template (accessed 2022-05-06)
[6] https://perlmaven.com/boolean-values-in-perl (accessed 2022-05-06)

visit data to the template, which creates rows for each symptom.
The usage of HTML::Template for the import function will be explained in Chapter 3.1.2.

2.3.1 SAMS Database

The SAMS database is separated into two main schemas: `sams_data` (Figure 5.9) and `sams_userdata` (Figures 5.11 and 5.10).
The user data (including private data like e-mail, password and role-specific data as well as visit data) is stored separately from SAMS data (HPO, OMIM and Orphanet) to ensure convenient export of the SAMS data without worrying about data security. A full overview of the relevant schemas for this thesis can be found in Figure 5.12. As one can see, these two schemas are connected only by the respective visit and corresponding visit data.
A Primary Key (PK) identifies a single record in a table and can be referred to by a Foreign Key (FK).
Essentially, tables are linked by the use of an FK with the ability to fuse different table if they have matching keys.
That way the data can be broken up into smaller, more manageable tables that can be easily joined by matching the PK of one table with the FK of other tables.

A selection of Perl basics facilitating the understanding of source code and Perl data structures can be found in Appendix 5.1.

2.4 Phenopacket components used in SAMS

In the following paragraphs I present a list of Phenopacket elements (”building blocks”) that were used in the import and export functions. These are a subset of all possible Phenopacket elements references in Chapter 1.3.

`subject`

The `subject` of the phenopacket, in most cases a patient or proband of a clinical study, is represented by an *Individual* element.
SAMS uses the `string id (REQUIRED)` (`string` is the data type, a concatenation of characters) and the `Sex sex (OPTIONAL)`. For a full list please visit the Phenopacket Documentation.

`ontologyClass`

The *OntologyClass* element is used to represent terms from ontologies as the HPO. Consisting of only two elements, a CURIE format `id` like `HP:1234567` and a `string label`, it finds its way into different corners of the phenopacket.
The Phenopacket standard requires that the `id` and the `label` match in the original ontology and a `MetaData Resource`.

`timeElement`

A *TimeElement* can take on different forms, for our use case in SAMS I chose to use either

- `Timestamp`,
- `TimeInterval` or
- `OntologyClass`

`Timestamp` is an ISO-8601 date string in the format
`{yyyy}-{mm}-{dd}T{hh}:{mm}:{ss}[.{frac_sec}]Z`
with optional fractional seconds and `"Z"` indicating the Coordinated Universal Time (UTC) timezone.

A `TimeInterval` consists of two `Timestamps`: `start` and `end`.

The `TimeElement OntologyClass` is used for the `onset` of `phenotypicFeatures` and `diseases`. Terms can be chosen from the HPO Onset subontology.

`phenotypicFeatures`

Phenotype in this context is referred to as "[a]n individual phenotypic feature, observed as either present or absent (excluded), with possible onset, modifiers and frequency"[7]. SAMS uses the `phenotypicFeature` for HPO terms with the `OntologyClass type (REQUIRED)` and optionally `TimeElement onset / resolution` as well as `bool excluded` (`true` if phenotype explicitly absent).

`diseases`

Analogous to `phenotypicFeature` with the difference of using OMIM and Orphanet as `terms`.

[7] Phenopacktes documentation for `phenotypicFeature`

`metaData`

The *MetaData* element contains `REQUIRED` parameters `Timestamp created`, `string createdBy` and `string phenopacketSchemaVersion`. A list of *Resources* has to be provided with one *Resource* per ontology or database used.

`resources`

A *Resource* element has to contain all of the following `REQUIRED` fields:

- `string id`: resource identifier e.g. `"hp"`
- `string name`: formal name e.g. `"human phenotype ontology"`
- `string namespacePrefix`: namespace prefix e.g. `"HP"`
- `string url`: namespace prefix e.g. `"http://www.human-phenotype-ontology.org"`
- `string version`: namespace prefix e.g. `"2022-02-22"`
- `string iriPrefix`: namespace prefix e.g. `"http://purl.obolibrary.org/obo/HP_"`

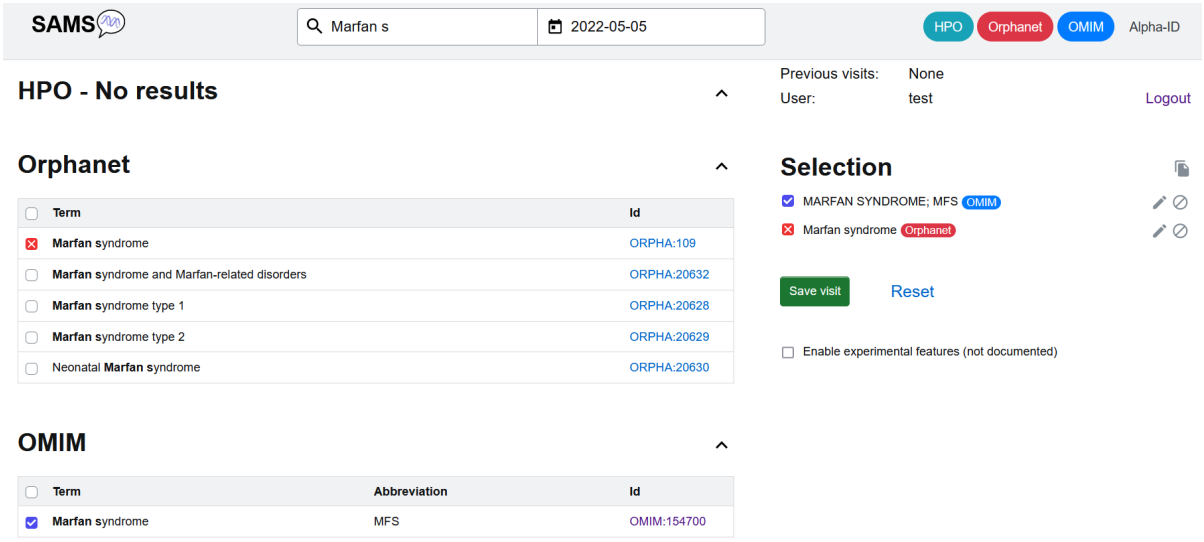


Figure 2.2: SAMS' *Enter phenotype* interface, searching for *"Marfan s"*.

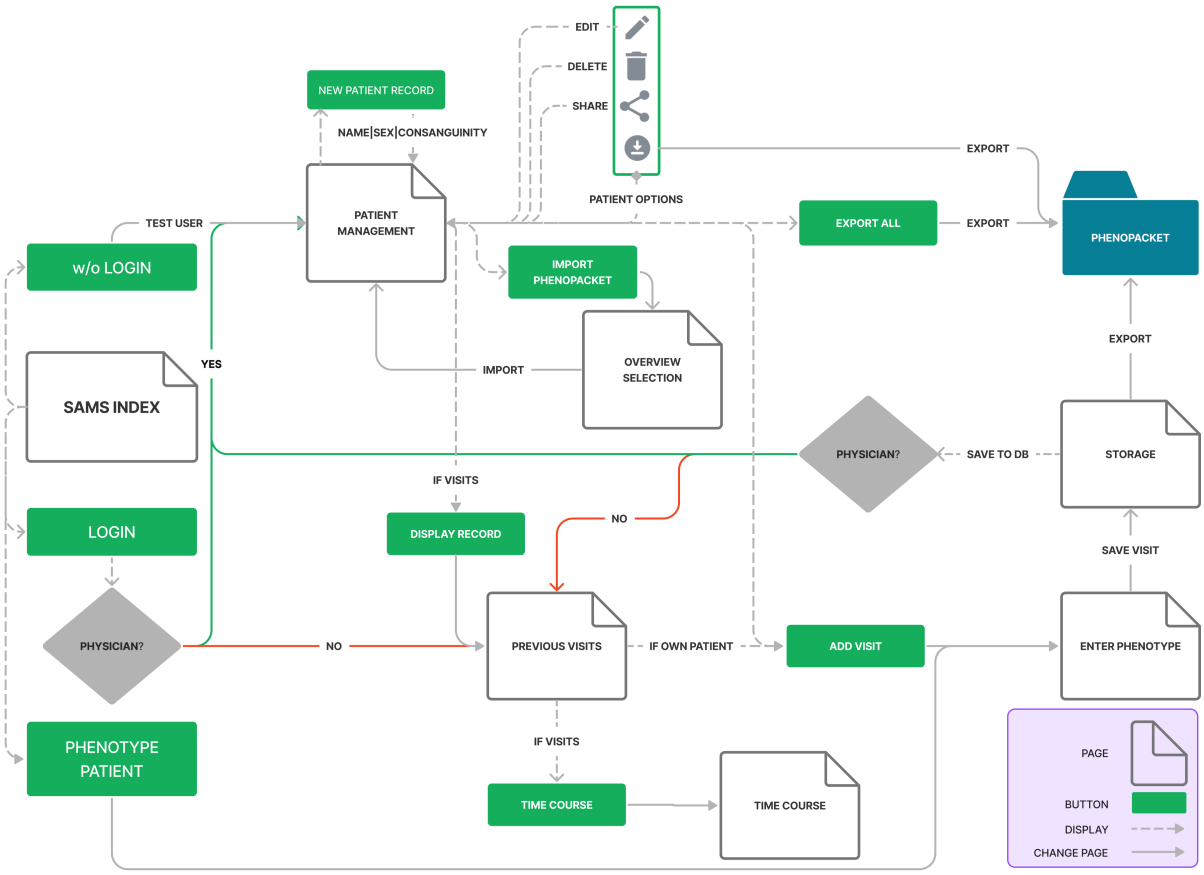


Figure 2.3: Page flow of the SAMS application.

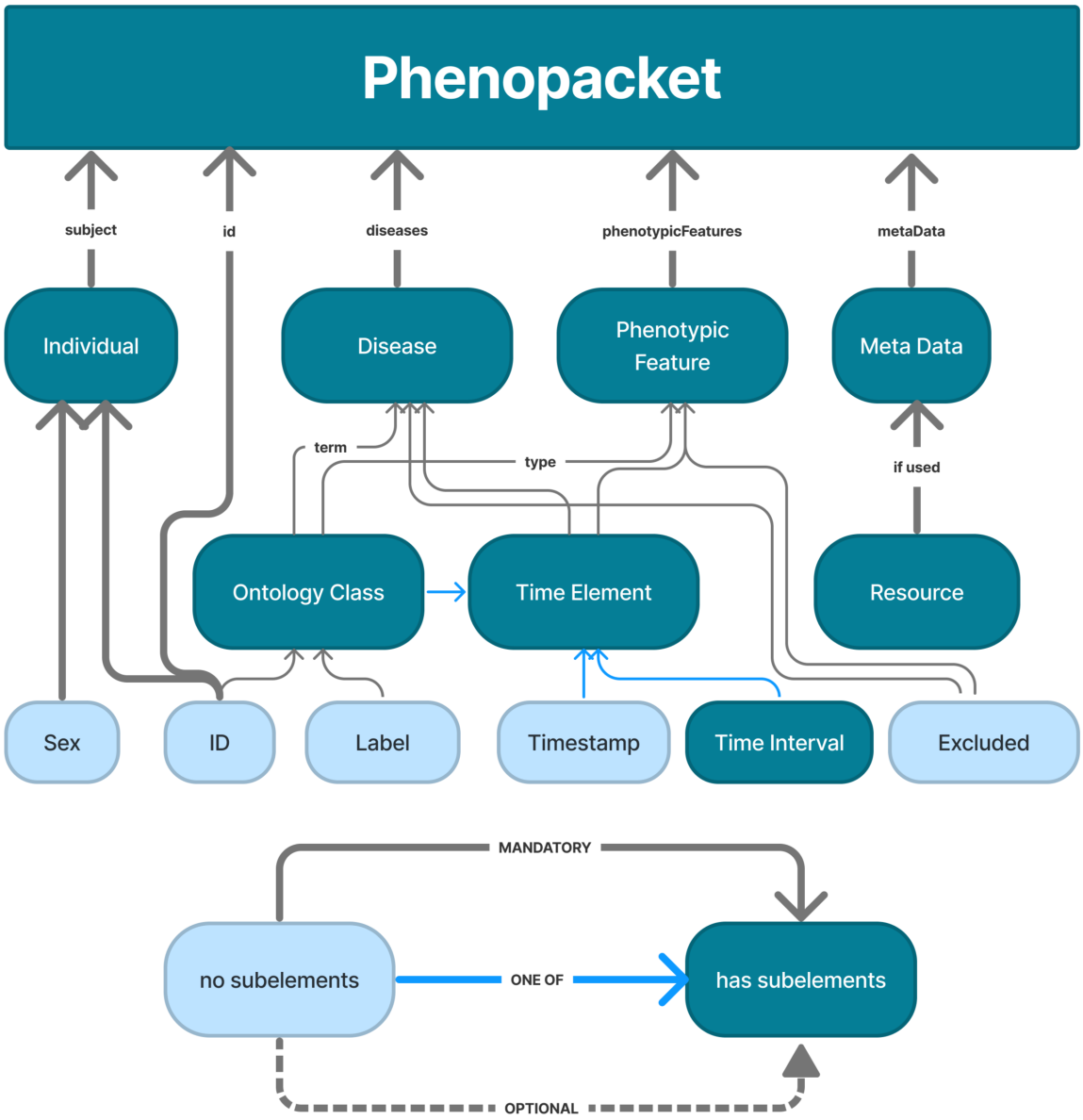


Figure 2.4: Phenopacket building blocks used in this thesis in SAMS.

3 Methods and Results

3.1 Novel SAMS functions

Source code of the SAMS application can be found in the SAMS GitLab repository[1]. The most recent version of my work is at the time of writing in branch `import_overview`[2] but will presumably be merged into the `devel` branch soon. Besides testing and debugging SAMS' page flow and functions, I implemented functions to read from and write to a Phenopacket as part of this thesis.
I will structure this chapter to emulate the circular data flow of figure 3.1, beginning from the left-hand side with the import from a JSON Phenopacket, continue with the internal handling and manipulation of data in SAMS and finish with the export of said data.

3.1.1 Demonstration Phenopacket

I will use the Marfan syndrome as an example for this thesis, as discussed in Chapter 1.0.1. Peter N. Robinson provided SAMS with multiple Phenopacket examples, which can be found in this GitHub repository.
The JSON file `marfan.json` contains the female patient *proband C*, last encountered at the age of 27.
One `disease` was annotated: Marfan syndrome (OMIM:154700). A `medicalAction` documents the oral intake of 30 milligrams of drug *losartan* twice daily from `2019-03-20` to `2021-03-20`.
I modified this Phenopacket to additionally include phenotypic abnormalities from the HPO. Using the frequencies provided by the HPO at https://hpo.jax.org/app/browse/disease/OMIM:154700, I chose *HP:0002616 Aortic root aneurysm*(45/58) and *HP:0030961 Microspherophakia*(2/53) [Nayak et al., 2021].
In addition, I added *HP:0004421 Elevated systolic blood pressure* (chosen on the basis of [Tehrani et al., 2021]) and a general term *HP:0032441 Blood group AB* with congenital onset.

[1] https://git.bihealth.org/btg/software/sams/sams
[2] https://git.bihealth.org/btg/software/sams/sams/-/tree/import_overview

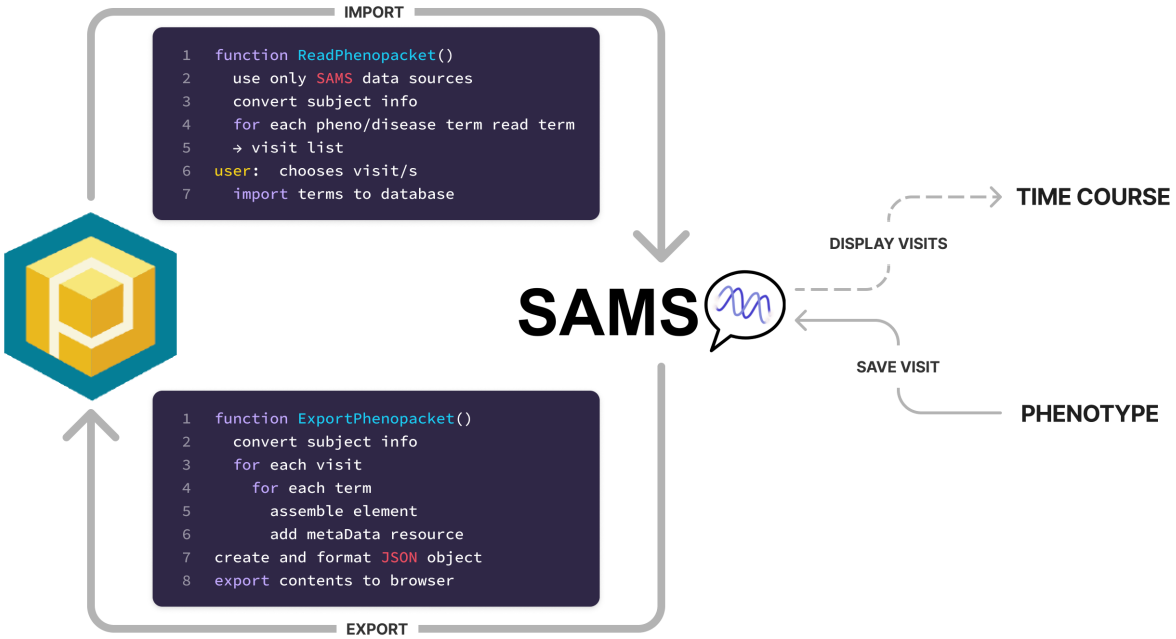


Figure 3.1: Data flow between Phenopackets and SAMS. Pseudo code summarizes the import and export procedure. Solid arrows symbolize data flow, the dashed arrow symbolizes the visualization of data.

To showcase different use cases, I constructed these four `phenotypicFeatures` with different combinations:

1. present with timestamp onset
2. present with congenital onset
3. absent with no onset

```
"type": {
  "id": "HP:0004421",
  "label": "Elevated systolic blood pressure"
},
"onset": {
  "timestamp": "2019-03-18T00:00:00Z"
}
```

Listing 2: `phenotypicFeature` *Elevated systolic blood pressure*, observed on `2019-03-18`

```
"type": {
  "id": "HP:0032441",
  "label": "Blood group AB"
},
"onset": {
  "ontologyClass": {
    "id": "HP:0003577",
    "label": "Congenital onset"
  }
}
```

Listing 3: `phenotypicFeature` *Blood group AB*, congenital onset

```
"type": {
  "id": "HP:0002616",
  "label": "Aortic root aneurysm"
},
"onset": {
  "timestamp": "2019-03-19T00:00:00Z"
}
```

Listing 4: `phenotypicFeature` *Aortic root aneurysm*, observed on `2019-03-19`

```
"type": {
  "id": "HP:0030961",
  "label": "Microspherophakia"
},
"excluded": true
```

Listing 5: `excluded phenotypicFeature` *Microspherophakia*, no time specified

Additionally, I modified the resources by incorporating resource items for HPO and OMIM while omitting resources for `medicalActions` to save unnecessary lines of code. This is technically not a valid Phenopacket, but SAMS ignores these regardless.
The JSON file is accessible in the GitHub repository for this thesis[3], I will not show it in its entirety in this thesis. Please refer to Figure 1.4 if feeling confused about the structure.

3.1.2 Phenopacket Import

The Phenopacket import mainly serves the flow of data into the SAMS application. Users can share their patients with SAMS' share function, so no Phenopacket is necessary to convey

[3] https://github.com/floherzler/SAMS-phenopackets-thesis

information to other SAMS users.

This brings forth the main goal of the import: To conserve as much information as possible and store it in SAMS' database.

At the beginning of my thesis, I started working on the import function `ReadPhenopacket()`, which reads the JSON Phenopacket. This function was intermittently modified by Prof. Dr. Seelow; a screenshot of the live SAMS version with the current import page can be seen in Figure 5.2.

Users may not want to import all of the visits extracted from a Phenopacket, so a part of my work was to implement a way for users to select and unselect visits, extending Prof. Dr. Seelow's work that he did based on my Phenopacket parsing.

The first limitation is that SAMS can currently only handle terms from the HPO, OMIM and Orphanet, so other terms are omitted and the user is warned with a message displayed on the overview page.

If a `subject` is provided, SAMS reads the `REQUIRED id` and `OPTIONAL sex`.

SAMS does not store the date of birth or age of patients, so even though it will most likely be present in many Phenopackets, I ignore it here.

Moving on to the next building block of a Phenopacket, the list of `phenotypicFeatures` is parsed if present. Empty Perl `hashes` for storing `used_dates`, `intervals`, `times` `tamps` and `no_time` elements are initialized to be filled with items later on.

Unsupported data sources are recognized by their prefix (supported sources are `HP`, `OMIM` and `ORPHA`) and the corresponding entries are skipped.

For every supported feature, the `status` and `id` are read, which are `REQUIRED` for each *OntologyClass*.

If the `excluded` parameter is provided, SAMS converts `true` or `false` to `absent` or `present`, respectively. In case of a missing `excluded` element, SAMS assumes the default value, which is `false` meaning the feature was observed and not excluded.

There are multiple possibilities supported by SAMS for handling the onset element:

1. timestamp
2. interval
3. timestamp + resolution timestamp
4. ontologyClass
5. no onset

A timestamped onset is handled by formatting the UTC timestamp to `'yyyy-mm-dd'` and saving it alongside the status and `hpo_modifiers` to the `%timestamp` hash under the key of the corresponding phenotype ID. This hash is structured as seen in Code 6.

```
my %timestamp = {
  'HP:1234567' => {
    '2022-04-15' => {
      'status' => 'present',
      'hpo_modifiers' => ['HP:7654321', 'HP:1234321']
    }
  },
  'HP:0000007' => {
    '2022-01-01' => {
      'status' => 'present'
    },
    '2022-01-02' => {
      'status' => 'absent'
    },
  }
}
```

Listing 6: Perl hash for example timestamps.

Intervals are saved to the `%interval` hash with following structure, using the `start` and `end` fields of the phenotypicFeature (Code 7).

```
my %interval = {
  'HP:1234567' => [
    '2022-01-01',
    '2022-01-02',
    'present'
  ]
}
```

Listing 7: Perl hash for an example interval.

For onset and resolution combined, SAMS uses the dates as beginning and end dates of an interval and assigns them to the `%interval` hash. This currently only works when both onset and resolution are `timestamps`.

***OntologyClass* onset elements** add an `hpo_modifier` to the timestamp, as displayed in Code 6. Currently only *Congenital onset* is supported, the internal representation for that is a visit date of `'1900-01-01'`.

If no onset is provided, SAMS emits a warning for the user, eliminating the possible error of a provided `resolution` without an `onset`.
The date is added to the `%used_dates` hash for all onset types except the `no_time` element, which stores the date in the `%no_time` hash.
Having now parsed all of the phenotypicFeatures, the procedure for the diseases is quite similar. Apart from using `term` instead of `type` as the name of the *OntologyClass*, it is the same.

After proceeding with all supported `phenotypicFeatures` and `diseases`, I now split all of them into timestamps to store as entries in the database.
Using nested loops, all of the visits of a patient that happened in the time span of that interval get assigned a timestamp for that feature or disease. For `no_time` elements, all of the visits are assigned a timestamp, excluding terms where a congenital onset applies, as they are also represented as visits internally (visit data `'1900-01-01'`).

Next, the `%timestamp` hash gets split into a list for the visits, iteratively assigning all of the timestamp to either a new or an existing visit. The resulting array for the visits `symps_arr` now gets passed by `import_phenopacket.cgi` to the `HTMPL::Template` `pheno_table.tmpl`, a tabular display used on multiple SAMS pages, accompanied by HTML warnings and messages for the user about the parsed elements.

The HTML::Template then uses the array of visits to construct a table of visits followed below by info messages for the user.

The user can un-select and re-select visits by clicking a checkbox next to every visit, calling the JavaScript function `change_checkbox_state()` from `some.js`, which changes the state of the HTML checkbox to a blue check mark if previously a red cross and vice versa (Screenshot 3.2 is set to import all three visits).
After being satisfied and clicking 'Import to DB', a function in JavaScript parses all the selected visits and modifies them, so they can be used in Perl.
The JavaScript function `selected_visits()` from `some.js` then selects all checked checkbox elements, which each have the value of their visit data (a date and a list of IDs and their statuses) and processes each selected visit to ultimately create a JavaScript object of all selected visits.

The template `pheno_table.tmpl`, filled with values by `import_phenopacket.cgi` now calls `import_phenopacket.cgi` again, but this time with the `imported_visits` in JavaScript Object Notation (JSON) from function `selected_visits()`.
Reading them into the Perl CGI script again is achieved by using the `JSON::Parse parse_json()`[4] function. This script now creates a new patient and calls the function `InsertNewVisit()` for each of the imported visits, which creates a new visit with all of the corresponding phenotypes and diseases.

`CreatePatient()` checks if the patient already exists and inserts a new patient into `sams_userdata.patients` if they do not, returning the incremented number.

`InsertNewVisit()` inserts data into `sams_userdata.visits` and calls `InsertPhenotype()` with the number of the new visit. HPO modifiers are currently only supported for HPO terms because of missing database work.

`InsertPhenotype()` inserts phenotypes into the corresponding database table. HPO modifiers are inserted into a new table `sams_userdata_visits_mxn_hpo`, representing an *m to n* relationship between the HPO terms and the `term_visit_numbers`[5]. This new table can be seen in Figure 5.10 and consists of only these two FKs.
An overview of all new database entries resulting from the import of the example Phenopacket can be seen in Figure 3.4. SAMS' time course for the newly created patient is displayed in Screenshot 3.3.

3.1.3 Phenopacket Export

The Phenopacket export enables users to conveniently export their patients' data in JSON format. The user is provided a download icon next to the patient record. By clicking the icon a new browser tab opens and displays the JSON code. The browser Firefox automatically opens a JSON viewer with highlighted syntax in which you can collapse or expand sections[6].
Additionally, users can export all of their patients with a button at the bottom of the page.

[4] https://metacpan.org/pod/JSON::Parse#parse_json (accessed 2022-05-06)
[5] a term can have multiple modifiers and an HPO modifier can be linked to many `term_visit_numbers`
[6] https://firefox-source-docs.mozilla.org/devtools-user/json_viewer/index.html (accessed 2022-05-06)

Individual export

By clicking the export icon for a particular patient, the `ExportPhenopacket()` function in `SAMS.pm` is called with the associated patient number. The function queries the database for the id, sex and visit data (without HPO modifiers) of this patient.
It converts the sex (e.g. from 'w' to 'FEMALE), the status from 'present' or 'absent' to `true` or `false` and creates a UTC timestamp from the saved date in 'yyyy-mm-dd' format by adding a `"T00:00:00Z"` to the date, marking it as the beginning of the visit day because SAMS does not store more accurate timestamps.
HPO phenotypes get assigned to the `phenotypicFeatures` list while a predefined resource element with meta data about the used HPO version gets added to the Phenopacket `metaData resources`.
ORPHA and OMIM terms are assigned to the `diseases` and the `resources` are added accordingly.

Finally, a Perl hash is built and returned from the function. This hash now is indented and manipulated by the `JSON::Create`[7] Perl module.
Booleans `true` and `false` are converted from Perl's `1` or `0` and the Phenopacket keys are ordered to always be consistent with the order suggested by the Phenopackets documentation because Perl hashes are not ordered and can vary between iterations. I defined a custom order of levels with increasing indices. When the `JSON::Create cmp()` function compares elements, it now compares the two indices that I defined and can determine which Phenopacket element appears first (Code 12 in the appendix).

Collective export

When exporting all the patients of a user, the `ExportPhenopacket()` function is simply called for every patient with their respective patient number. The resulting list of Phenopackets is then exported analogous to the individual export.

`On-the-fly` export

After creating a visit, the user is prompted to choose between saving the visit to the database or exporting it as a phenopacket. This export mode works in a slightly different way than the other two, because it doesn't query the database (the visit is not stored in there yet). It takes the data from the just now created visit and passes it to `ReadPhenopacket()`.

[7] https://metacpan.org/pod/JSON::Create (accessed 2022-05-06)

The remaining JSON processing steps are identical.

Resulting Phenopacket

The resulting Phenopacket can be found in this thesis' GitHub repository [8].
The `id (arbitrary.id)` is the same for every created Phenopacket, while the `subject` of the Phenopacket in this case is defined by `"id": "proband "` and `"sex": "FEMALE"`.
For every entry in the database (as seen in Figure 3.4 tables for HPO and OMIM) `Export Phenopacket()` created either a `phenotypicFeature` or a `disease` with the `onset` being the timestamp from SAMS database. HPO modifiers are not exported.
The meta data of the Phenopacket is constructed using the timestamp of the export, deriving it from the server using Perl's `gmtime()` function[9], the used resources with a placeholder for the version and the e-mail of the user who created the Phenopacket for the `createdBy` field.

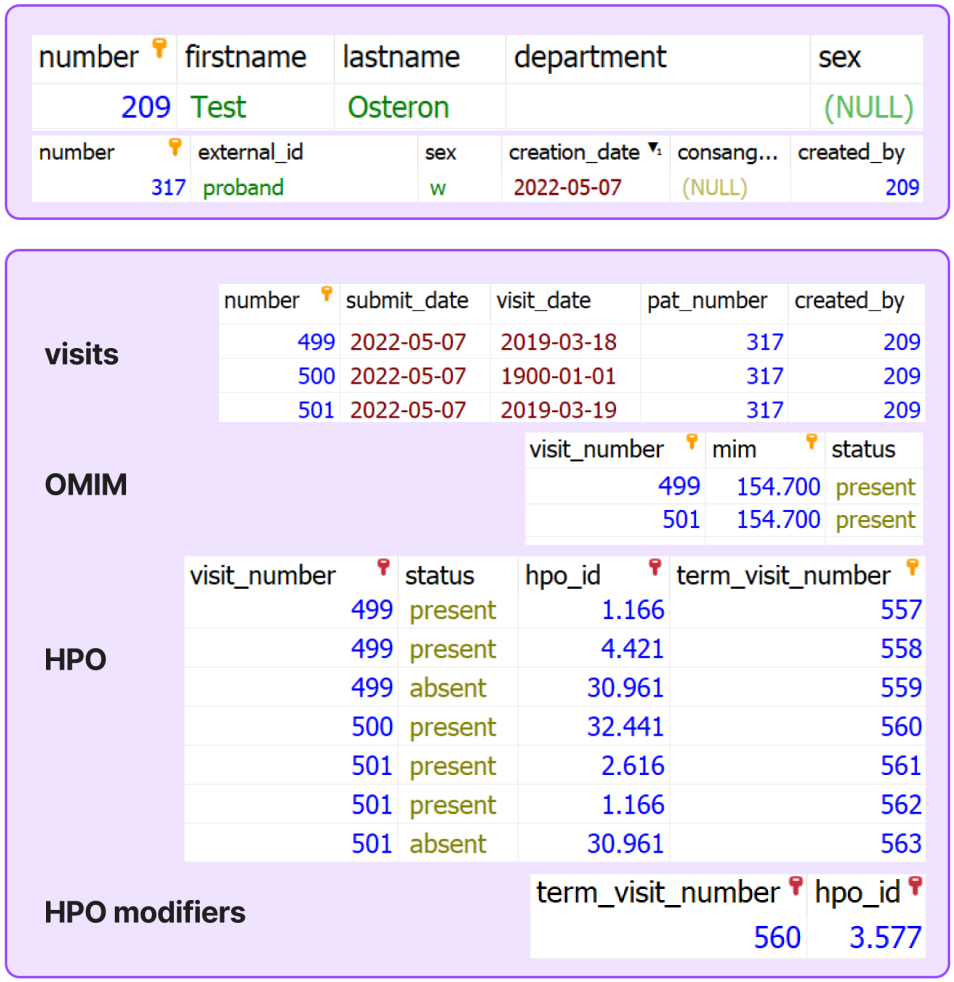
[8] https://github.com/floherzler/SAMS-phenopackets-thesis
[9] https://perldoc.perl.org/functions/gmtime (accessed 2022-05-08)

Visit date	Records	IDs	Import
2019-03-18			☑
	☒ Microspherophakia	HP:0030961	
	☑ Marfan syndrome	OMIM:154700	
	☑ Elevated systolic blood pressure	HP:0004421	
	☑ Arachnodactyly	HP:0001166	
1900-01-01			☑
	☑ Blood group AB	HP:0032441	
2019-03-19			☑
	☒ Microspherophakia	HP:0030961	
	☑ Marfan syndrome	OMIM:154700	
	☑ Arachnodactyly	HP:0001166	
	☑ Aortic root aneurysm	HP:0002616	

Figure 3.2: Import preview for Marfan syndrome and HPO `phenotypicFeatures`.

Data source	ID	Term	2019-03-18	2019-03-19	congenital
HPO	HP:0002616	Aortic root aneurysm		present	
HPO	HP:0001166	Arachnodactyly	present		
HPO	HP:0032441	Blood group AB			present
HPO	HP:0004421	Elevated systolic blood pressure	present		
HPO	HP:0030961	Microspherophakia	absent		
OMIM	OMIM:154700	MARFAN SYNDROME; MFS	present		

Figure 3.3: SAMS time course for the imported Marfan syndrome Phenopacket.



number	firstname	lastname	department	sex
209	Test	Osteron		(NULL)

number	external_id	sex	creation_date	consang...	created_by
317	proband	w	2022-05-07	(NULL)	209

visits

number	submit_date	visit_date	pat_number	created_by
499	2022-05-07	2019-03-18	317	209
500	2022-05-07	1900-01-01	317	209
501	2022-05-07	2019-03-19	317	209

OMIM

visit_number	mim	status
499	154.700	present
501	154.700	present

HPO

visit_number	status	hpo_id	term_visit_number
499	present	1.166	557
499	present	4.421	558
499	absent	30.961	559
500	present	32.441	560
501	present	2.616	561
501	present	1.166	562
501	absent	30.961	563

HPO modifiers

term_visit_number	hpo_id
560	3.577

Figure 3.4: All database entries resulting from the import of the Marfan syndrome Phenopacket.
The upper section are the doctor with internal number `209` and the created new patient *proband* with internal number `317`.
Three visits are created and referenced from the `visits_mxn_omim` and `visits_mxn_hpo` tables.
`visits_mxn_hpo_modifiers` references the latter for an HPO modifier.

4 Discussion

In this work I demonstrated the import and export of GA4GH Phenopackets with data from the SAMS database. I implemented functions to parse a JSON phenopacket and manipulate the data to ultimately import it to the database. Additionally, I implemented a function to reverse this process and exported SAMS' patient data as Phenopackets in JSON.
While being a successful project, there are certain aspects that can be improved.
I will discuss them in this final chapter of the thesis, divided into improvements for SAMS and shortcomings of my work.

4.1 Improvements for SAMS

During my work on the import and export functionality I discovered some aspects of SAMS that could be improved.

***Clinical modifiers*,** as found in the subontology of the HPO, are not separated from *Phenotypic abnormalities* in SAMS' phenotyping UI.
A patient can therefore be phenotyped with the symptoms *Congenital onset* (HP:0003577) and *Moderate (Severity)* (HP:0012826) which medically do not make sense on their own. As seen in Figure 1.2, the HPO separated these top-level subontologies conveniently. Therefore I suggest to exclude *Clinical modifiers, Mode of inheritance* and *Frequency* from the search terms on `enter_phenotype.cgi` in the future. Thus the resulting HPO terms are exclusively medically relevant *Phenotypic abnormality, Blood group* and *Past medical history*.
The excluded subontologies could then be used to refine phenotyping via an additional option to add HPO `modifiers` to a `phenotypicFeature` in the `modifier` element[1].

The ORPHA-ID displayed while searching for diseases with SAMS and viewing the time course is intended to link to the corresponding entry on www.orpha.net (e.g. https://www.orpha.net/consor/cgi-bin/OC_Exp.php?Lng=EN&Expert=558 for `ORPHA:558` *Marfan syndrome*).

[1] https://phenopacket-schema.readthedocs.io/en/latest/phenotype.html#modifiers (accessed 2022-05-08)

As seen in Figure 4.1, SAMS stores two different numeric terms for a disease (`disorder_id` and `orpha_number`), deriving them from the official XML (Extensible Markup Language) file that Orphanet provides at www.orphadata.org[2].
Unfortunately, SAMS displays the `disorder_id` when it should be displaying the `orpha_number`. Following the link redirects the user to the correct disease, albeit with a different URL in the browser search bar. SAMS' target group are medical professionals and the user experience has to be as smooth as possible. An incorrect ORPHA code (ORPHA:558 for *Marfan syndrome*) surely confuses medical professionals and possibly disrupts their diagnosis work flow.

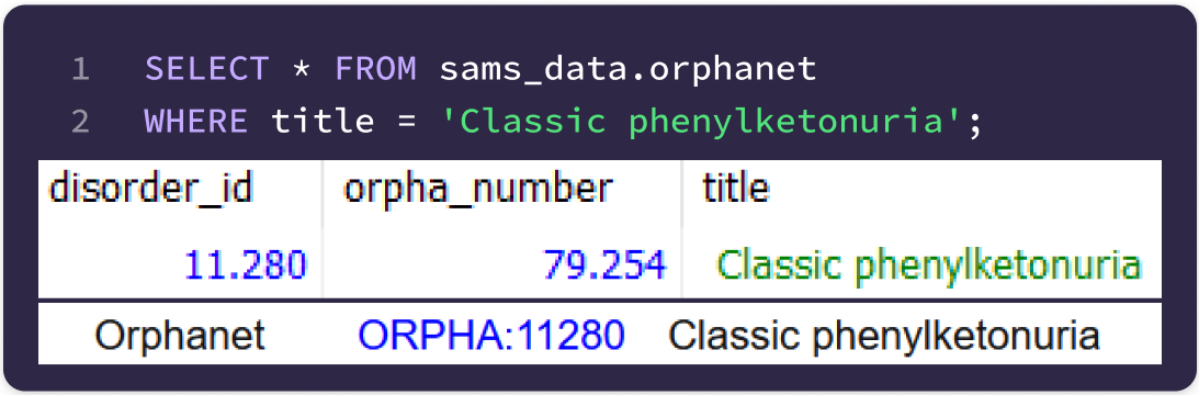


Figure 4.1: SQL query for database entry *Classic PKU* and display in the SAMS time course.

This issue was noted by the SAMS team and will soon be fixed in order to comply with the typically used ORPHA IDs.

4.2 Shortcomings of my work

The Phenopackets documentation states: *"While it is possible to inter-operate with other services using JSON produced from hand-crafted/alternative implementations, we strongly suggest using the schema to compile any required language implementations"*[3]. The Phenopacket Schema provides tools for Java, Python and C++[4].
As the task for this thesis was to integrate the import and export functions into the SAMS application, the decision was made to use Perl for JSON manipulation. This resulted in easy manipulation of SAMS' data without relying on the Phenopacket tools.

[2] e.g. http://www.orphadata.org/data/xml/en_product1.xml (accessed 2022-05-02)
[3] https://phenopacket-schema.readthedocs.io/en/latest/working.html (accessed 2022-05-06)
[4] Phenopacket Schema GitHub

Phenopackets can be encoded in JSON or YAML. Both file types are human and machine-readable, with YAML being used in the Phenopackets documentation (less brackets, see Figure 1.4) and JSON needing more lines of code for the same information. Nevertheless, JSON was chosen as the fitting data type for SAMS on the basis of being practical for data exchange, especially between web pages while YAML is more often used for configuration files.

4.2.1 Phenopacket import

HPO modifiers were introduced in the later stages of this thesis and therefore have quite some unfinished tasks. The foundation for importing modifiers from a Phenopacket is laid, but so far only HP:0003577 Congenital onset can be imported, other modifiers are skipped. The HPO `phenotypicFeature` element has a `modifiers` element to allow a list of modifiers from the HPO or other ontologies. The `disease` element does not have a similar element and can only be modified by specifying the `onset` and `resolution` as well as the `disease_stage`.

Displaying modifiers can be improved significantly. Currently, a 'congenital' column (Figure 3.3) is displayed identically to columns for other visit dates. If a term has been given the modifier, an additional box is displayed in the right-most column. That is counter intuitive, as one would suspect a genetic disease that was already present at birth to also be displayed chronologically **before** other visits that of course took place after birth of the patient.

Interpreting missing `disease` or `phenotypicFeature` `timeElements` can be ambiguous, as without medical knowledge a program cannot safely make assumptions about the most likely option.

Phenotypic abnormalities without an `onset` parameter are difficult to categorize because of the diverse range that the HPO covers, blood group should be interpreted differently from fever.

Diseases without an `onset` parameter might have been present at birth, but without further knowledge of the disease no program can simply assume that, as this parameter may have been forgotten by the creator of the Phenopacket. A medical doctor may assume it to be obvious that a certain disease can only be of *congenital onset*, but SAMS needs to know additional information to present it in an intuitive, user-friendly manner.

We made the decision for SAMS to display these phenotypes and diseases for all the visits of a patient (excluding *1900-01-01 - congenital onset*), as seen in Chapter 3.1.2.

An option is to show the disease or symptom in the list but instead of a bar labeled with the `status`, one could display a more fitting message: **“no time specified” (Figure 3.3)**.

Age and date of birth are not saved to SAMS’ database and will therefore be ignored for the foreseeable future of SAMS.

Intervals are not saved as such in SAMS explicitly, but rather as a series of time stamps (mentioned in Chapter 3.1.2). This leads to loss of information, as SAMS only knows that a disease was present at the start and end of the original Phenopacket `interval` and possible intermittent visits. More detailed time stamps than the day are ignored by SAMS (equivalent to being stores as `"T00:00:00Z"`). I will discuss the implications of this in Chapter 4.2.2.

Import layout could be improved by changing the use of the same checkbox design for different purposes. Currently, the SAMS checkbox is used for representing a `present` or `absent` feature as well as for a selected or unselected visit. Users might be confused by that and try to unselect different symptoms within a visit, which is not supported yet. The imported visits should be ordered by descending date, so the most recent visit is at the top and *congenital onset* (instead of *1900-01-01*) at the bottom of the list.

HPO parents should be checked before importing, as omitting these checks can lead to medical contradictions. The standard SAMS phenotyping process emits a warning for an absent HPO parent when the HPO child is present(“HPO 3368 is marked as absent - this is impossible because it is a parent.”) and vice versa (“HPO 3368 is marked as present - this is not necessary because it is a parent.”) as seen in Figure 4.2.

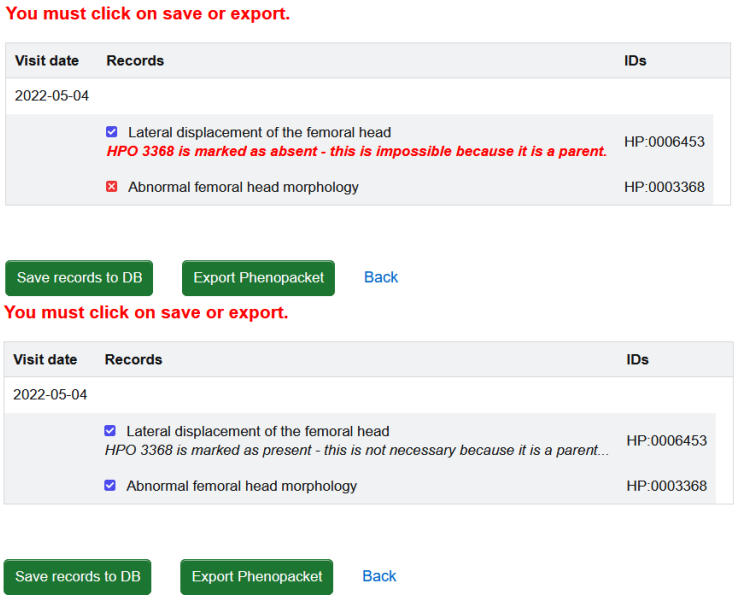


Figure 4.2: Screenshot of SAMS - `phenotype2db.cgi`.

4.2.2 Phenopacket Export

Certainly the biggest weakness of this thesis is the missing official validation of the resulting Phenopacket. I studied the Phenopackets documentation in its entirety and am therefore confident that Phenopackets created by my export function follow all of the rules of the Phenopacket schema. That said, I cannot completely rule out any edge cases where my export function will do unexpected things.

An official Phenopacket validator is available on GitHub in the `phenopacket-tools` directory, but because of my unfamiliarity with Java and the `Maven` build process, this was not achieved in the time frame of this thesis. The created JSON has been tested for correctness using the online tool JSONlint. Both individual and all-patient export have passed the test but the Phenopacket validator is needed to verify the resulting JSON files from the Phenopackets perspective.

All-patient Phenopacket export is currently solved by exporting a list of Phenopackets (Code 8).

```
[
    {
        "subject":{...},
        "phenotypicFeatures": [...],
        "diseases": [...],
        "metaData" : {...}
    },
    {
        "subject":{...},
        "phenotypicFeatures": [...]
        "diseases": [...],
        "metaData" : {...}
    }
]
```

Listing 8: Schematic export of multiple patients.

This is valid JSON. However, it is not suggested by the Phenopacket Schema, which proposes the top-level elements listed in Chapter 1.3. The usage of the `Family` element can be useful in other cases but not to export all patients, as they in all likelihood are unrelated. The `Cohort` element is designed to bundle study participants or subjects that are connected in a medically relevant way. It was initially ruled out due to the missing clinical link between potential co-patients. The Phenopacket Schema is missing a top-level element to simply group unrelated Phenopackets, which is why I decided on exporting Phenopackets as a list.

SAMS does not feature an import for a list of Phenopackets, so regardless of `Cohort` or list, SAMS is as of yet unable to handle multiple Phenopackets at once.

Exporting intervals is not implemented, as mentioned in Chapter 3.1.3. SAMS stores visits, at which a `disease` or `phenotypicFeature` can be either `present` or `absent` and exports a Phenopacket block for every visit date. This way the resulting Phenopacket conserves all of the information from SAMS' database, but as seen in Chapter 3.1.3, listing a disease twice with an `onset` element for each is not possible for a genetic disease and has to be separated with a `resolution` for non-genetic, transient diseases.

Exporting HPO modifiers is not supported by SAMS yet but can be added to the database retrieval of visit dates and added to the `phenotypicFeature modifiers` list.

For the sake of user friendliness, the export's file name suggested by the browser should always be JSON. Firefox offers a **Save** button, but the suggested filename then is `export_phenopacket.c / export_all_phenopacket.cgi.json`. The JSON file can be opened normally.

In Google Chrome, single-patient export and subsequent saving correctly saves it as `export_phenopack` All-patient export is saves as a CGI file. This is certainly confusing for the user.

It was explicitly requested to export it as a separate page with the text, so I did not implement a direct export from Perl.

4.3 Outlook

With SAMS' mode for patients, it is feasible to incorporate more supported Phenopacket structures. A pedigree[5] can be entered by patients themselves, gathering more relevant data without any additional time of medical personnel. Subjects can also be bundled in a family[6] element.

SAMS' team is working on implementations of further annotation systems, the Mondo Disease Ontology (MONDO) being next.

Currently only available in English, SAMS will be made available in German soon, but the development is hindered by the German HPO translation.

Alpha-IDs are, as mentioned in the Introduction, incorporated into SAMS but currently not fully developed. They do not show up in the phenotyping result list on the development server due to a missing update.
Alpha-ID-SE codes will be mandatory for rare disease encoding if both ICD and Orphanet code are present from 2023 onwards due to German law "Digitale–Versorgung–und–Pflege–Modernisierungs–Gesetz (DVPMG)" (Link to official PDF, page 115).
More work has to be done to include Alpha-ID-SEs fully into SAMS as I could only find a fraction of terms when searching the official SAMS web page. Because the IDs are specific to German users, the entire SAMS page simultaneously has to be translated to German.

A differential diagnosis guide in SAMS is currently in development, suggesting a diagnosis based on entered phenotypes.

Before SAMS is to ultimately be embedded in hospital information systems, these issues have to be addressed. With the integration of Global Alliance For Genomics and Health Phenopackets, SAMS is one step closer to becoming adopted by the medical community.

[5] Phenopackets Documentation: Family
[6] Phenopackets Documentation: Family

5 Appendix

Figures

SAMS pages

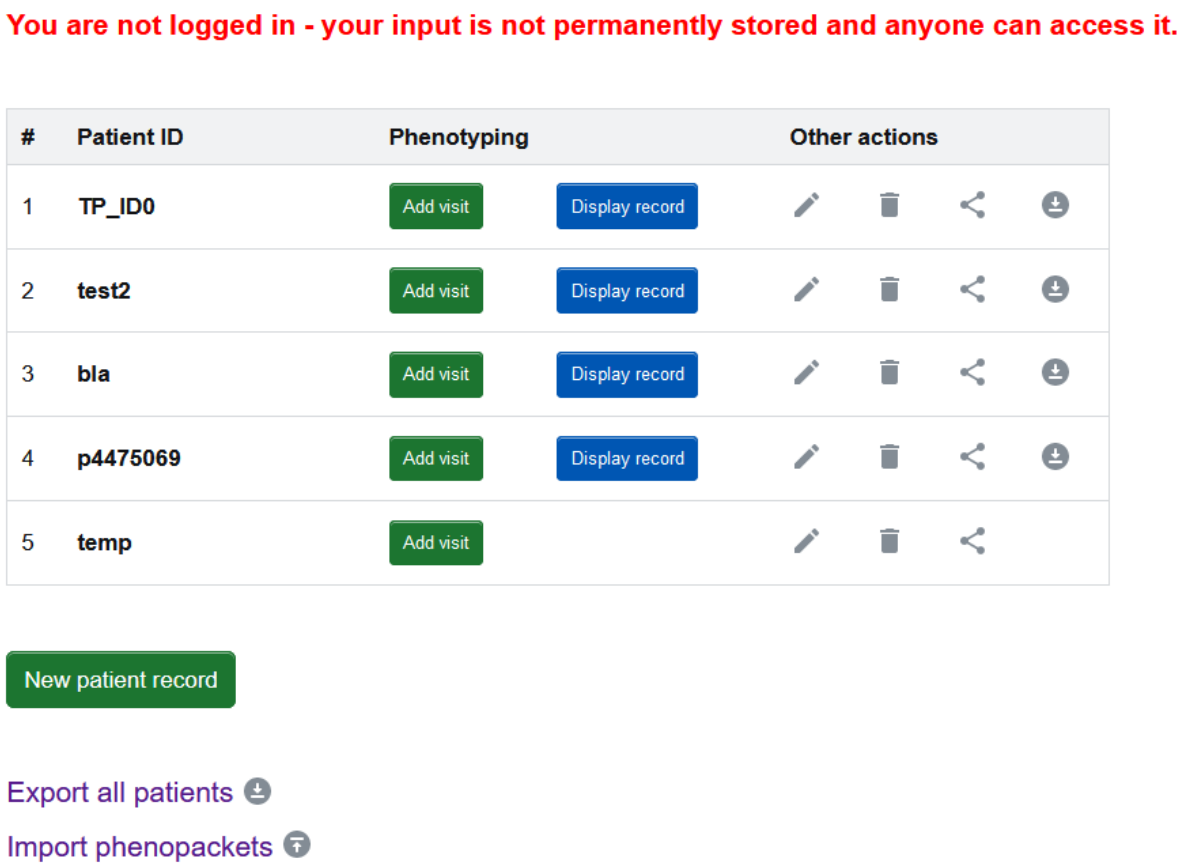


Figure 5.1: SAMS without logging in.

PATIENT INSERTED.

This is a phenopacket for allSrcImport (FEMALE) (created by test@osteron.de on 20220424)
NO TIME HP:0031264 present Abnormal Bowman capsule morphology
TIMESTAMP HP:0002027 present Abdominal pain 20220101
NO TIME ORPHA:14479 present Farmer's lung disease
TIMESTAMP OMIM:609803 present ANKYRIN AND ARMADILLO REPEATS-CONTAINING PROTEIN; ANKAR 20220217
TIMESTAMP OMIM:609803 present ANKYRIN AND ARMADILLO REPEATS-CONTAINING PROTEIN; ANKAR 20220201
TIMESTAMP OMIM:616569 present CYSTEINE SULFINIC ACID DECARBOXYLASE; CSAD 20220102
USED DATES:
20220101, 20220102, 20220201, 20220217, 20220424
INSERT PATIENT allSrcIm (sex: f)
original ID allSrcImport was longer than 8 characters, truncated to allSrcIm
create VISIT on 20220101 as 694
create VISIT on 20220102 as 695
create VISIT on 20220201 as 696
create VISIT on 20220217 as 697
create VISIT on 20220424 as 698
WARNINGS
SAMS cannot handle data from orphanet yet. These entries were skipped.
SAMS cannot handle data from hp yet. These entries were skipped.
SAMS cannot handle data from omim yet. These entries were skipped.
SAMS cannot handle metaData data yet - these entries were skipped.
Subject (patient) ID allSrcImport longer than 8 characters, truncated to allSrcIm

Go back

Figure 5.2: `Import Phenopacket` page as implemented by Prof. Dr. Seelow.

Screenshots Data Source Websites

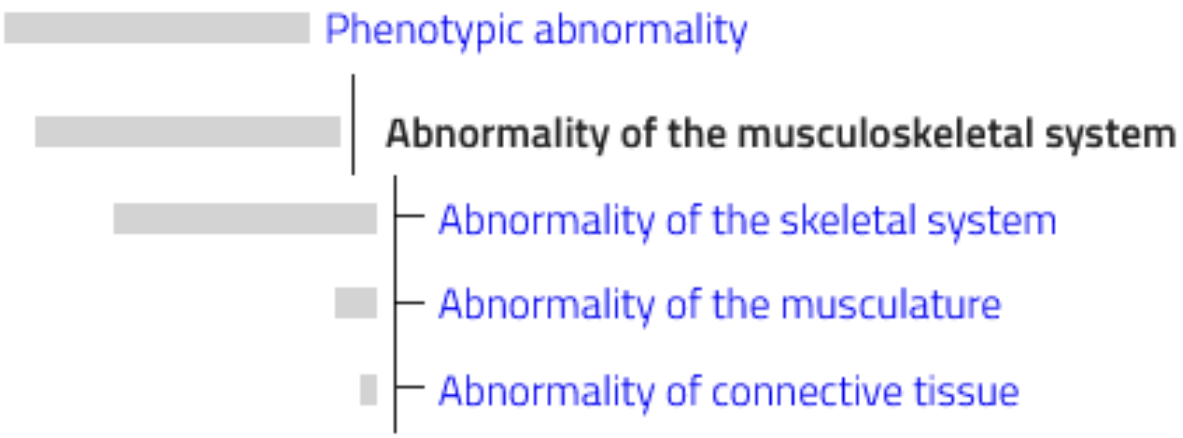
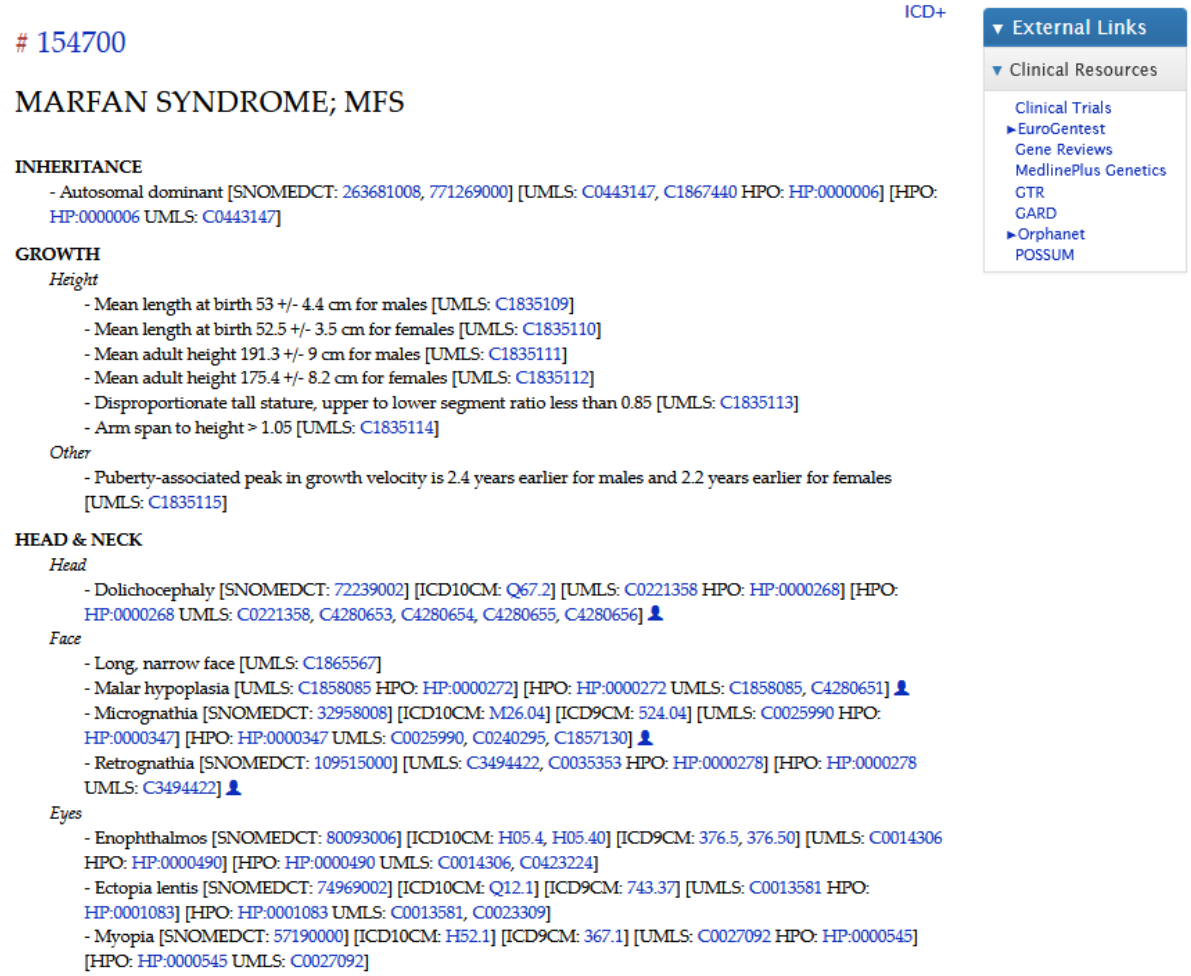


Figure 5.3: Screenshot of HPO *Abnormality of the musculoskeletal system* subontology.



ICD+

154700

MARFAN SYNDROME; MFS

INHERITANCE

- Autosomal dominant [SNOMEDCT: 263681008, 771269000] [UMLS: C0443147, C1867440 HPO: HP:0000006] [HPO: HP:0000006 UMLS: C0443147]

GROWTH

Height

- Mean length at birth 53 +/- 4.4 cm for males [UMLS: C1835109]
- Mean length at birth 52.5 +/- 3.5 cm for females [UMLS: C1835110]
- Mean adult height 191.3 +/- 9 cm for males [UMLS: C1835111]
- Mean adult height 175.4 +/- 8.2 cm for females [UMLS: C1835112]
- Disproportionate tall stature, upper to lower segment ratio less than 0.85 [UMLS: C1835113]
- Arm span to height > 1.05 [UMLS: C1835114]

Other

- Puberty-associated peak in growth velocity is 2.4 years earlier for males and 2.2 years earlier for females [UMLS: C1835115]

HEAD & NECK

Head

- Dolichocephaly [SNOMEDCT: 72239002] [ICD10CM: Q67.2] [UMLS: C0221358 HPO: HP:0000268] [HPO: HP:0000268 UMLS: C0221358, C4280653, C4280654, C4280655, C4280656]

Face

- Long, narrow face [UMLS: C1865567]
- Malar hypoplasia [UMLS: C1858085 HPO: HP:0000272] [HPO: HP:0000272 UMLS: C1858085, C4280651]
- Micrognathia [SNOMEDCT: 32958008] [ICD10CM: M26.04] [ICD9CM: 524.04] [UMLS: C0025990 HPO: HP:0000347] [HPO: HP:0000347 UMLS: C0025990, C0240295, C1857130]
- Retrognathia [SNOMEDCT: 109515000] [UMLS: C3494422, C0035353 HPO: HP:0000278] [HPO: HP:0000278 UMLS: C3494422]

Eyes

- Enophthalmos [SNOMEDCT: 80093006] [ICD10CM: H05.4, H05.40] [ICD9CM: 376.5, 376.50] [UMLS: C0014306 HPO: HP:0000490] [HPO: HP:0000490 UMLS: C0014306, C0423224]
- Ectopia lentis [SNOMEDCT: 74969002] [ICD10CM: Q12.1] [ICD9CM: 743.37] [UMLS: C0013581 HPO: HP:0001083] [HPO: HP:0001083 UMLS: C0013581, C0023309]
- Myopia [SNOMEDCT: 57190000] [ICD10CM: H52.1] [ICD9CM: 367.1] [UMLS: C0027092 HPO: HP:0000545] [HPO: HP:0000545 UMLS: C0027092]

▼ External Links

▼ Clinical Resources

Clinical Trials
►EuroGentest
Gene Reviews
MedlinePlus Genetics
GTR
GARD
►Orphanet
POSSUM

Figure 5.4: Screenshot of *Marfan syndrome - Clinical Synopsis* at OMIM.org.
References to HPO and Orphanet terms as well as ICD codes can be found in the list on the left and the *Clinical Resources* expandable list on the right.

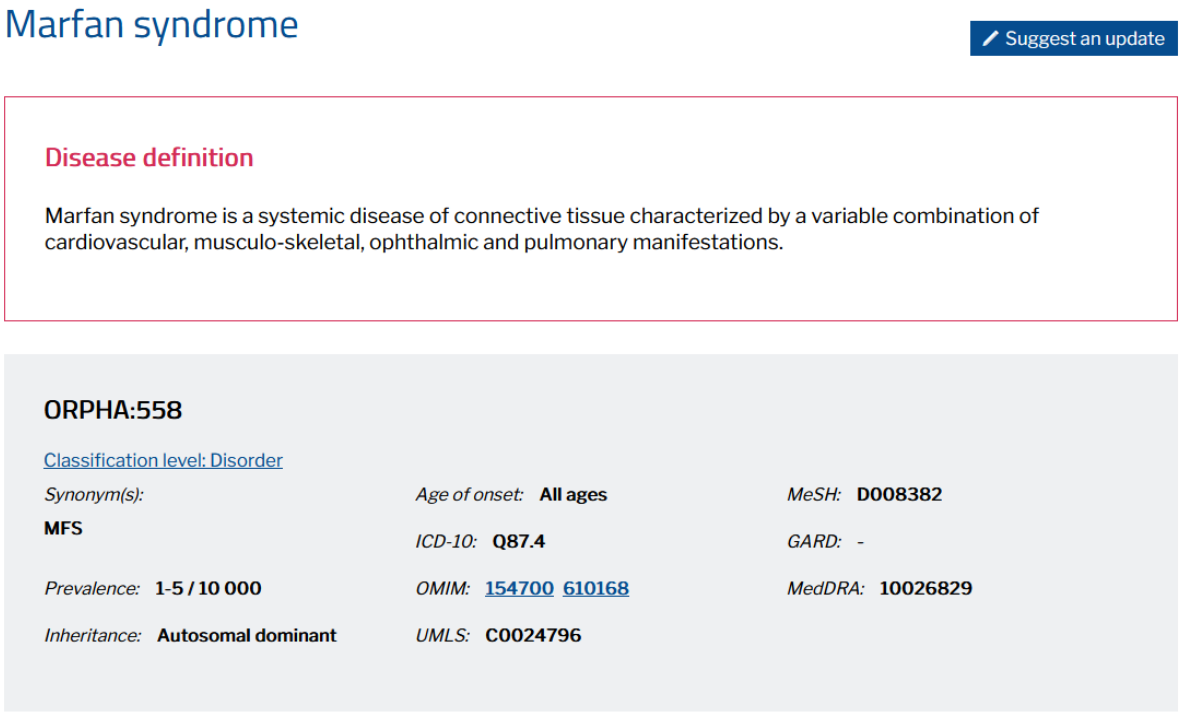


Figure 5.5: Screenshot of *Marfan syndrome* at orpha.net.

Figures from other authors

B. L. Loeys et al., 2010

Table 1 Features of differential diagnosis

Differential diagnosis	Gene	Discriminating features
Loeys–Dietz syndrome (LDS)	TGFBR1/2	Bifid uvula/cleft palate, arterial tortuosity, hypertelorism, diffuse aortic and arterial aneurysms, craniosynostosis, clubfoot, cervical spine instability, thin and velvety skin, easy bruising
Shprintzen–Goldberg syndrome (SGS)	FBN1 and other	Craniosynostosis, mental retardation
Congenital contractural arachnodactyly (CCA)	FBN2	Crumpled ears, contractures
Weill–Marchesani syndrome (WMS)	FBN1 and ADAMTS10	Microspherophakia, brachydactyly, joint stiffness
Ectopia lentis syndrome (ELS)	FBN1 LTBP2 ADAMTSL4	Lack of aortic root dilatation
Homocystinuria	CBS	Thrombosis, mental retardation
Familial thoracic aortic aneurysm syndrome (FTAA)	TGFBR1/2, ACTA2	Lack of Marfanoid skeletal features, levido reticularis, iris flocculi
FTAA with bicupid aortic valve (BAV)		
FTAA with patent ductus arteriosus (PDA)	MYH11	
Arterial tortuosity syndrome (ATS)	SLC2A10	Generalised arterial tortuosity, arterial stenosis, facial dysmorphism
Ehlers–Danlos syndromes (vascular, valvular, kyphoscoliotic type)	COL3A1, COL1A2, PLOD1	Middle sized artery aneurysm, severe valvular insufficiency, translucent skin, dystrophic scars, facial characteristics

Jacobsen et al., 2021

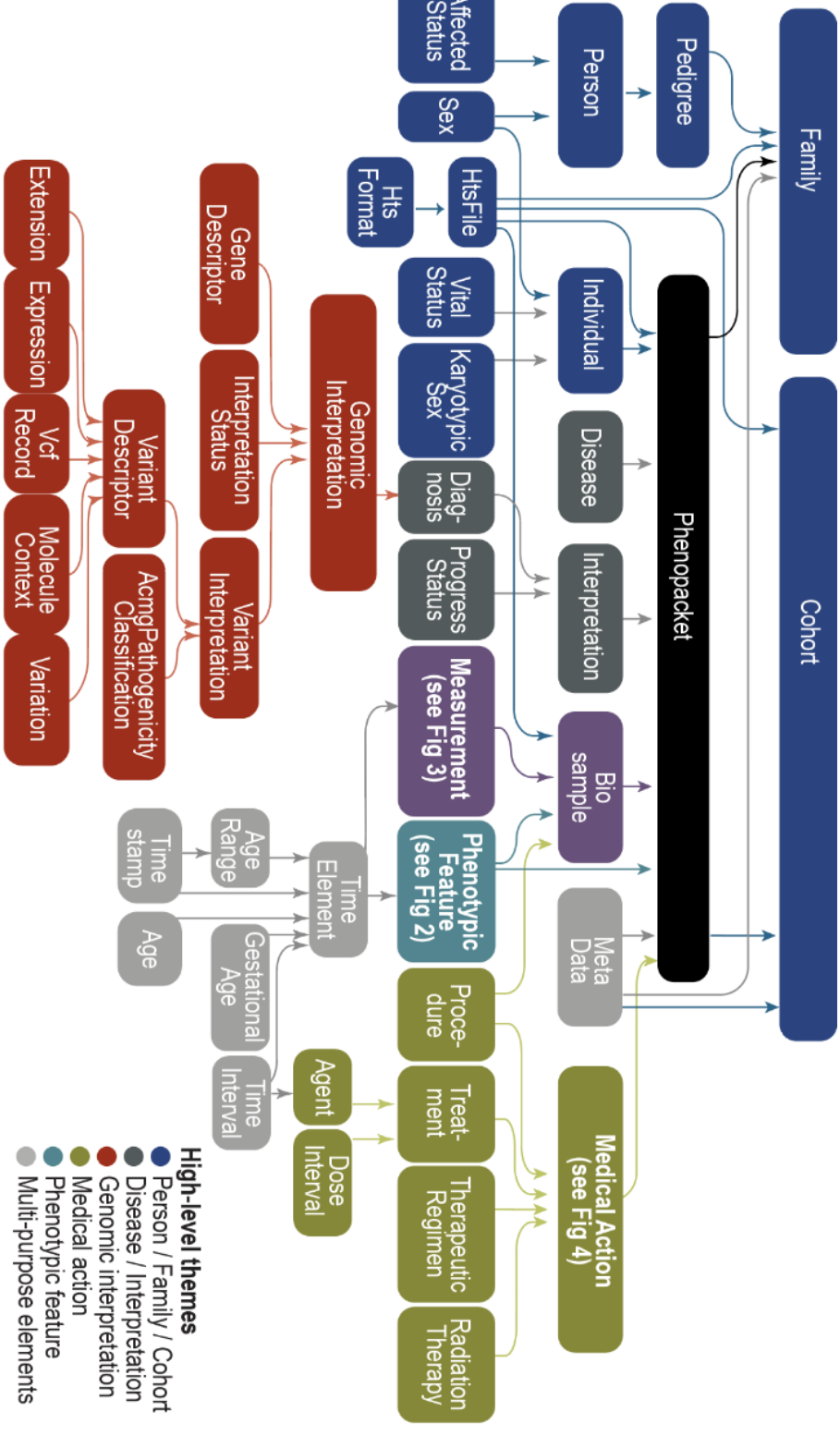


Figure 1. Phenopacket schema overview. *The GA4GH Phenopacket schema consists of several optional elements, each of which contains information about a certain topic such as phenotype, variant, pedigree, etc. One element can contain other elements, which allows a hierarchical representation of data. For instance, the Phenopacket contains elements of type Individual, PhenotypicFeature, Biosample, and so on. Individual elements can therefore be regarded as* ***building blocks*** *that are combined to create larger structures. Colors represent the major themes of elements within the schema. See Figures 2, 3, and 4 for specifications of the PhenotypicFeature, Measurement, and Medical Action elements.*

Figure 5.6: Overview over the elements in the GA4GH Phenopacket Schema.

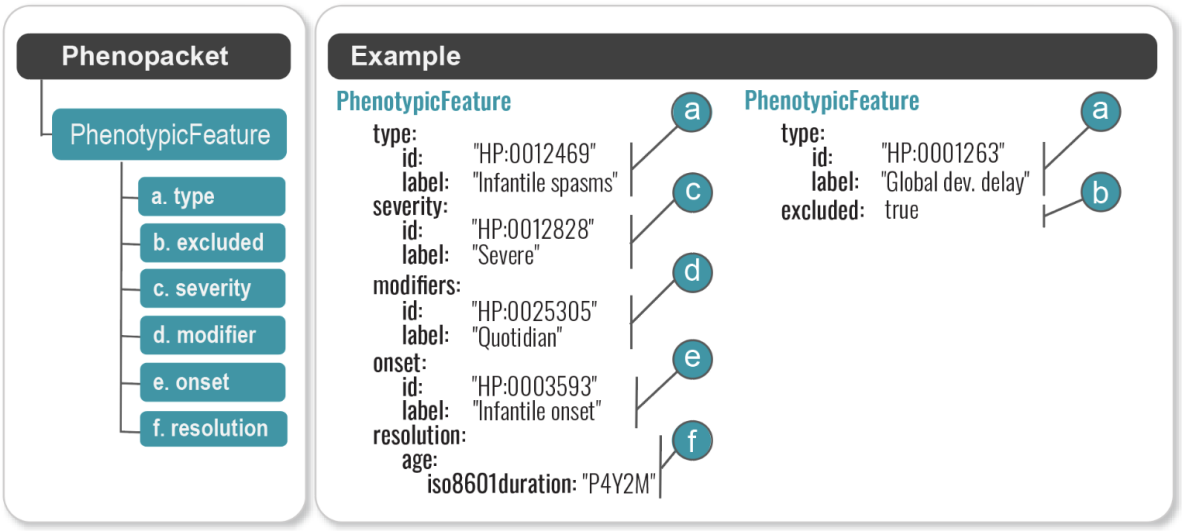


Figure 2. PhenotypicFeatures in a Phenopacket. *A Phenopacket can contain information about an arbitrary number of phenotypic features observed in a single individual, each encoded using a PhenotypicFeature element. For medical use cases the subject will generally be a patient or a proband of a study, and the phenotypes will be abnormalities described by an ontology such as the HPO. Each phenotypic feature is defined by an HPO term (a), which is qualified as either present or absent (excluded) (b), with possible severity (c), modifiers (d), onset (e), and resolution (f). The example in the right panel shows a phenotypic feature, severe daily infantile spasms, which first occurred in infancy and resolved at age 4 years and 2 months, in a child without global developmental delay.*

Figure 5.7: Elements of a `phenotypicFeature`.

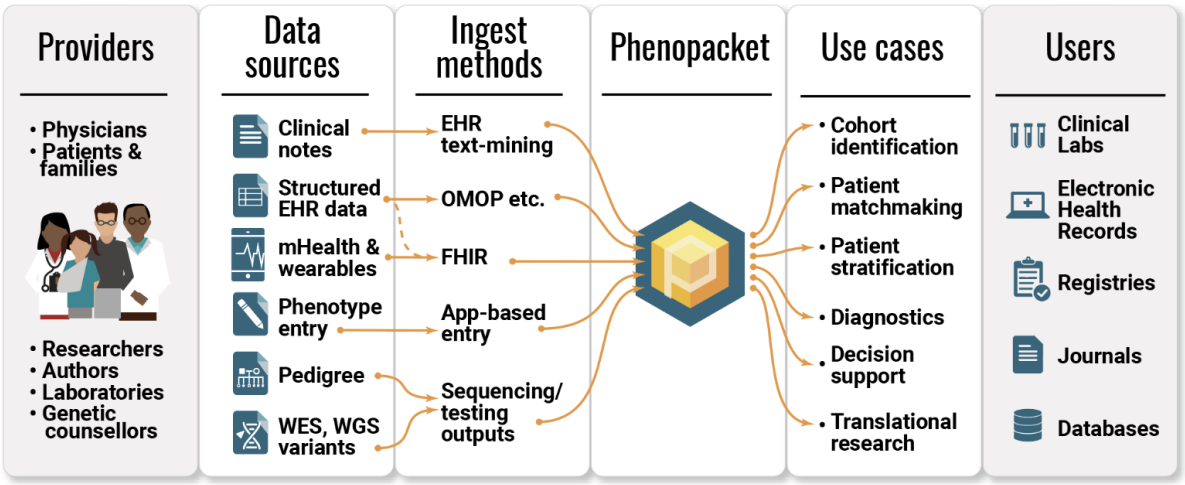


Figure 5. Phenotype data exchange in the biomedical ecosystem. *Multiple providers of phenotypic data include patients and clinicians, via a variety of mechanisms including mHealth and the EHR. The Phenopacket schema acts as a common model that can capture data from many sources with a unified software representation and in turn can be used by multiple receivers of the phenotypic information, including journals, databases, registries, clinical laboratories. Phenopackets can support diverse users and use cases, including patient matchmaking services, diagnostics, and cohort identification.*

Figure 5.8: Data exchange with the GA4GH Phenopacket Schema.

Tobias Schalau

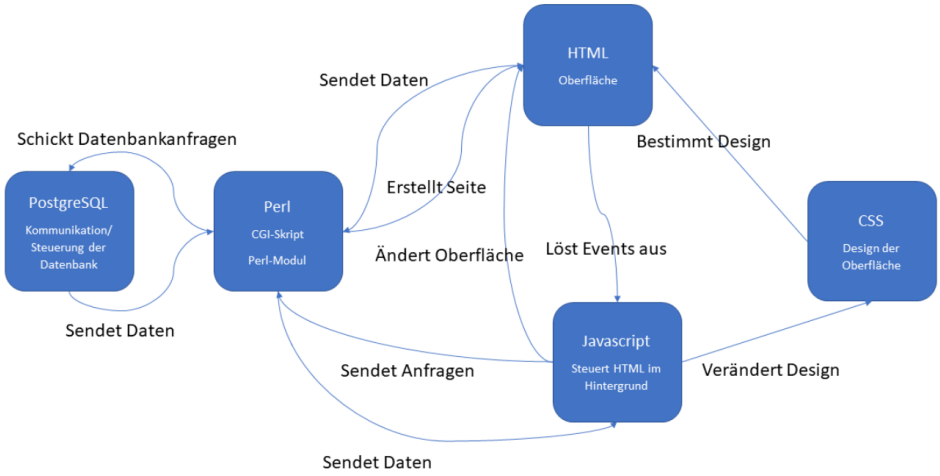


Abbildung 3: Die Grafik stellt den Zusammenhang zwischen den Programmiersprachen Perl, Javascript, des Datenbankmanagementsystems PostgeSQL und der Dokumentbeschreibung HTML, sowie dessen Stil CSS dar.

5.0.1 Tables

SAMS database

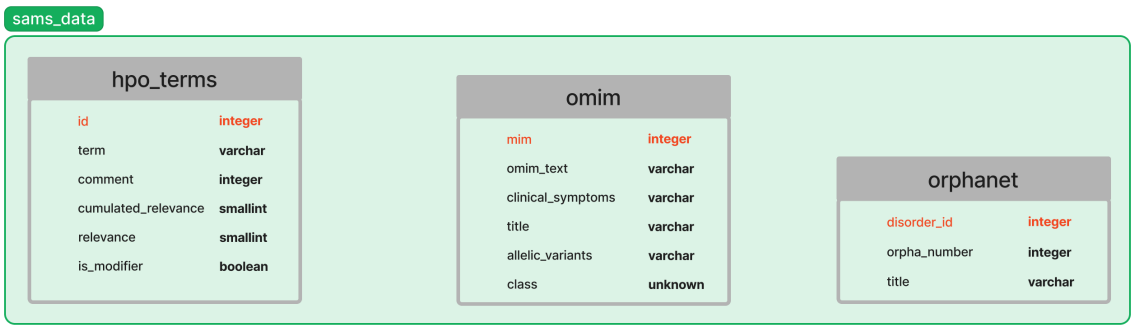


Figure 5.9: SAMS database schema `sams_data`.

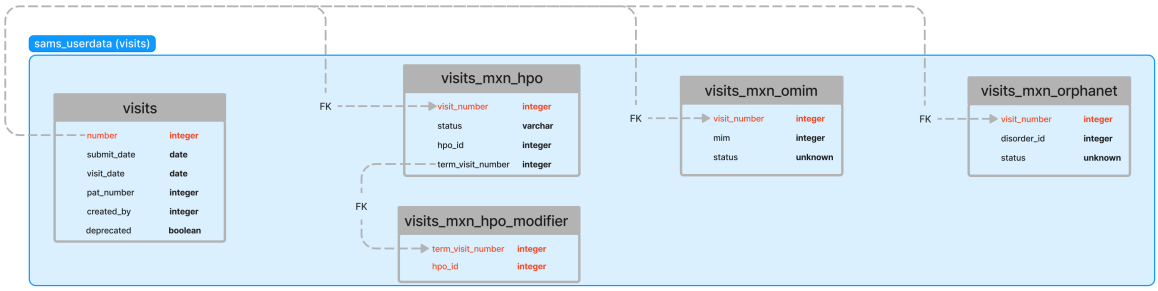


Figure 5.10: SAMS database schema `sams_userdata`, visits part.

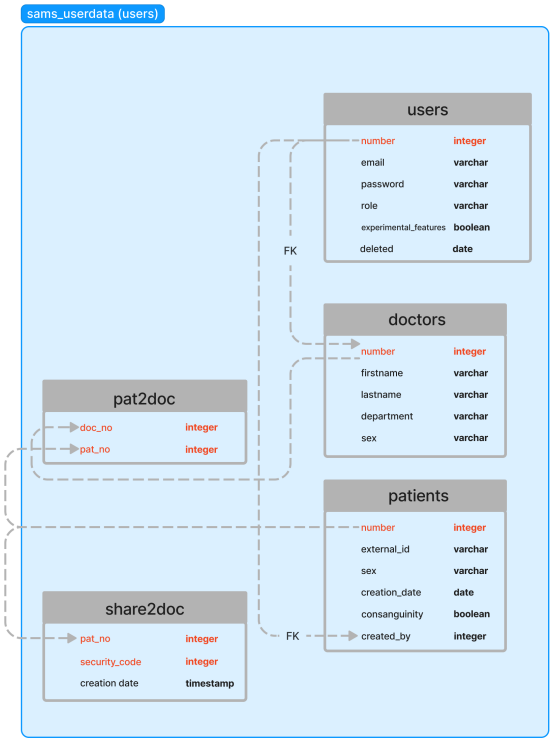


Figure 5.11: SAMS database schema `sams_userdata`, users part.

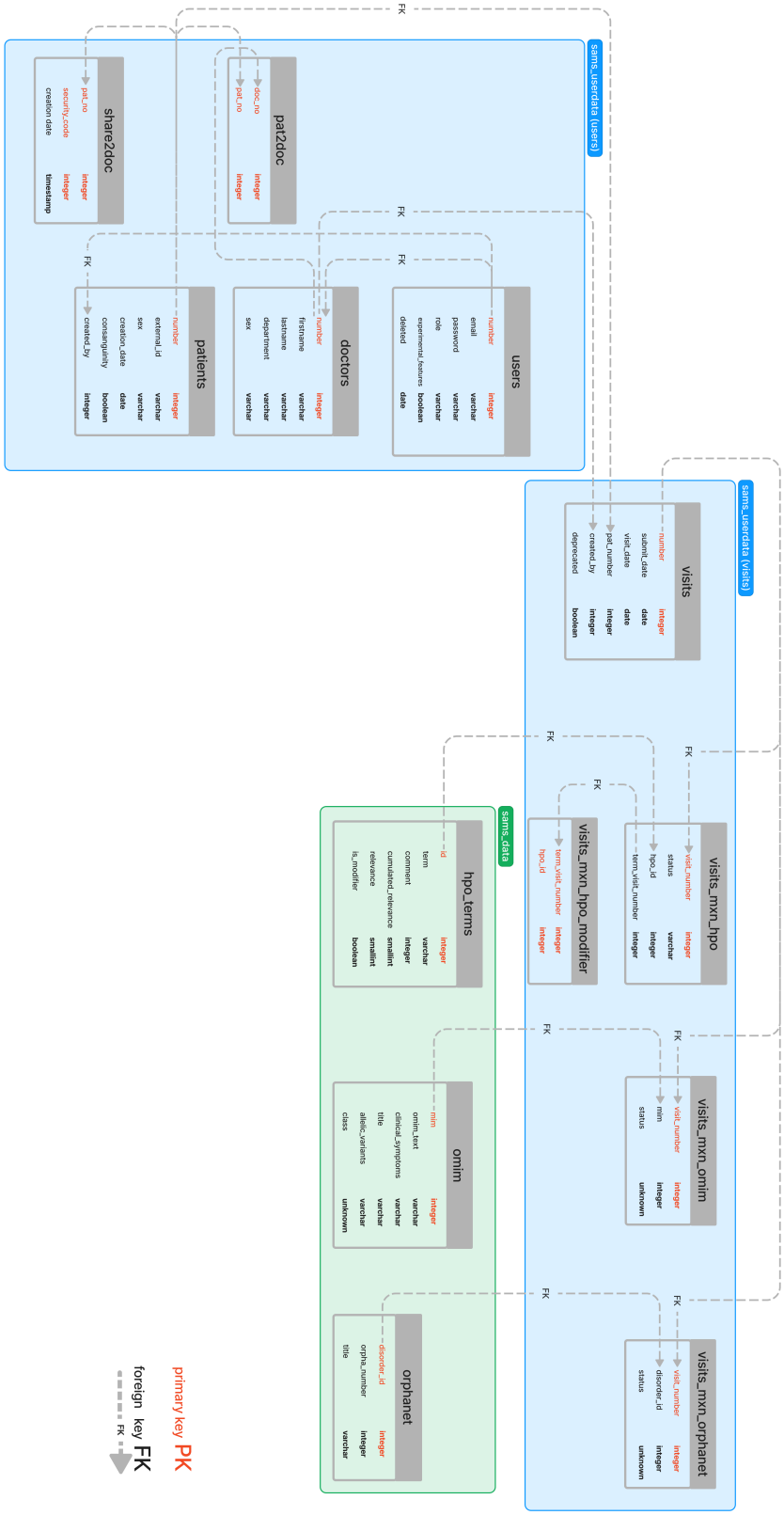


Figure 5.12: SAMS database schemas `sams_data` and `sams_userdata`, combined.

number	firstname	lastname	department	sex
209	Test	Osteron		(NULL)

number	external_id	sex	creation_date	consang...	created_by
317	proband	w	2022-05-07	(NULL)	209

visits

number	submit_date	visit_date	pat_number	created_by
499	2022-05-07	2019-03-18	317	209
500	2022-05-07	1900-01-01	317	209
501	2022-05-07	2019-03-19	317	209

OMIM

visit_number	mim	status
499	154.700	present
501	154.700	present

HPO

visit_number	status	hpo_id	term_visit_number
499	present	1.166	557
499	present	4.421	558
499	absent	30.961	559
500	present	32.441	560
501	present	2.616	561
501	present	1.166	562
501	absent	30.961	563

HPO modifiers

term_visit_number	hpo_id
560	3.577

Figure 5.13: Combined screenshots of the database when importing the Marfan syndrome Phenopacket.

5.1 Perl basics

Perl is used in SAMS because of its binding character between a web server and a database, emitting HTML pages.

For code readability, I will explain a few basic principles of Perl before going into detail about the resulting functions.

Three kinds of variables, declared with `my`[1] to only be accessible locally, were used in the source code of import and export functions:

- `$scalar`
- `@array`
- `%hash`

Scalars , prefixed with a `$`, do not have a `type`, they can contain `strings`, `numbers` (`integer` or `floating point`) or references other data.

```
my $string = "This is a string.";
my $integer = 42;
```

Listing 9: Perl scalars.

Arrays , prefixed with a `@`, are lists of scalars and can be accessed at a specific index as well as appended to.

```
my @array = [42, "This is a string."];
```

Listing 10: Perl array.

Hashes , prefixed with a `%`, differ from arrays in how data can be accessed. A hash is a list of `key - value` pairs, as shown in Code 11.

```
my %hash = {
    key1 => "value1",
    key2 => 2
};
```

Listing 11: Perl hash with two keys, `string` and `integer` values.

[1] Perldoc

Source code

```
sub OrderPhenopacketKeys {
    my ($elem1, $elem2) = @_;
# digits correspond to level, order from
# https://phenopacket-schema.readthedocs.io/en/latest/phenopacket.html
    my %elementOrder = (
        id => -1, #id always first
        subject => 1,
          sex => 11,
        phenotypicFeatures => 2,
        diseases => 3,
          type => 12,
          term => 13,
            label => 101,
          onset => 14,
          resolution => 15,
            timestamp => 102,
            interval => 103,
              start => 1001,
              end => 1002,
          excluded => 16,
        metaData => 4,
          created => 17, #metaData
          createdBy => 18,
          resources => 19,
            name => 104,
            url => 105,
            version => 106,
            namespacePrefix => 107,
            iriPrefix => 108,
          phenopacketSchemaVersion => 20);
    return $elementOrder{$elem1} > $elementOrder{$elem2} ? 1 : -1;
}
```

Listing 12: OrderPhenopacketKeys.

Table 5.1: Exported HPO terms for `OMIM:154700` *Marfan syndrome*.

HPO_TERM_ID	HPO_TERM_NAME	CATEGORY
HP:0001166	Arachnodactyly	Limbs
HP:0003758	Reduced subcutaneous adipose tissue	Connective tissue
HP:0000098	Tall stature	Growth
HP:0001377	Limited elbow extension	Limbs
HP:0001371	Flexion contracture	Connective tissue
HP:0001382	Joint hypermobility	Skeletal system
HP:0000006	Autosomal dominant inheritance	Inheritance
HP:0002650	Scoliosis	Skeletal system
HP:0002647	Aortic dissection	Cardiovascular
HP:0002616	Aortic root aneurysm	Cardiovascular
HP:0000189	Narrow palate	Head and neck
HP:0007676	Hypoplasia of the iris	Eye
HP:0002751	Kyphoscoliosis	Skeletal system
HP:0003302	Spondylolisthesis	Skeletal system
HP:0002097	Emphysema	Respiratory System
HP:0030961	Microspherophakia	Eye
HP:0008132	Medial rotation of the medial malleolus	Limbs
HP:0008138	Equinus calcaneus	Limbs
HP:0002107	Pneumothorax	Respiratory System
HP:0004872	Incisional hernia	Connective tissue
HP:0100775	Dural ectasia	Nervous System
HP:0001065	Striae distensae	”Skin, Hair, and Nails”
HP:0001083	Ectopia lentis	Eye
HP:0004970	Ascending tubular aorta aneurysm	Cardiovascular
HP:0004927	Pulmonary artery dilatation	Cardiovascular
HP:0000678	Dental crowding	Head and neck
HP:0003088	Premature osteoarthritis	Skeletal system
HP:0000767	Pectus excavatum	Skeletal system
HP:0000768	Pectus carinatum	Skeletal system
HP:0012773	Reduced upper to lower segment ratio	Growth
HP:0003199	Decreased muscle mass	Musculature
HP:0003179	Protrusio acetabuli	Limbs
HP:0025599	Inferior oblique muscle overaction	Eye
HP:0025586	Hypertropia	Eye
HP:0000278	Retrognathia	Head and neck
HP:0000275	Narrow face	Head and neck
HP:0000276	Long face	Head and neck
HP:0000272	Malar flattening	Head and neck
HP:0000268	Dolichocephaly	Head and neck
HP:0005136	Mitral annular calcification	Cardiovascular
HP:0002816	Genu recurvatum	Limbs
HP:0000218	High palate	Head and neck
HP:0001519	Disproportionate tall stature	Growth

Table 5.2: Selected HPO annotations for OMIM:154700 (Marfan syndrome) from `genes_to_phenotype.txt`, classified as "very rare" by HP:0040284.

entrez gene id	entrez gene sym-bol	HPO Term ID	HPO Term Name	Raw	Frequency HPO	Additional Info from G-D source	G-D source	disease-ID for link
2200	FBN1	HP:0001166	Arachnodactyly	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0100775	Dural ectasia	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0008138	Equinus calca-neus	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0000767	Pectus excava-tum	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0001647	Bicuspid aortic valve	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0001763	Pes planus	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0000565	Esotropia	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0001704	Tricuspid valve prolapse	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0001065	Striae disten-sae	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0001761	Pes cavus	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0001635	Congestive heart failure	-	-	-	mim2gene	OMIM:154700
2200	FBN1	HP:0005136	Mitral annular calcification	-	-	-	mim2gene	OMIM:154700
2200	FBN1	HP:0005180	Tricuspid regurgitation	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0025586	Hypertropia	-	HP:0040284	-	mim2gene	OMIM:154700
2200	FBN1	HP:0000276	Long face	-	-	-	mim2gene	OMIM:154700

List of Figures

Bibliography

Loeys, B., Nuytinck, L., Delvaux, I., De Bie, S., & De Paepe, A. (2001). Genotype and phenotype analysis of 171 patients referred for molecular study of the fibrillin-1 gene FBN1 because of suspected Marfan syndrome. *Archives of Internal Medicine*, *161*(20), 2447–2454. https://doi.org/10.1001/archinte.161.20.2447

Rosenthal, E., Biesecker, L., & Biesecker, B. (2001). Parental attitudes toward a diagnosis in children with unidentified multiple congenital anomaly syndromes. *American Journal of Medical Genetics*, *103*(2), 106–114. https://doi.org/10.1002/ajmg.1527

EURORDIS. (2007). Survey of the delay in diagnosis for 8 rare diseases in Europe (EurordisCare2). Fact sheet EurordisCare 2, 2.

Robinson, P. N., Köhler, S., Bauer, S., Seelow, D., Horn, D., & Mundlos, S. (2008). The Human Phenotype Ontology: A tool for annotating and analyzing human hereditary disease. *American Journal of Human Genetics*, *83*(5), 610–615. https://doi.org/10.1016/j.ajhg.2008.09.017

Loeys, B. L., Dietz, H. C., Braverman, A. C., Callewaert, B. L., De Backer, J., Devereux, R. B., Hilhorst-Hofstee, Y., Jondeau, G., Faivre, L., Milewicz, D. M., Pyeritz, R. E., Sponseller, P. D., Wordsworth, P., & De Paepe, A. M. (2010). The revised Ghent nosology for the Marfan syndrome. *Journal of Medical Genetics*, *47*(7), 476–485. https://doi.org/10.1136/jmg.2009.072785

Radonic, T., de Witte, P., Baars, M. J. H., Zwinderman, A. H., Mulder, B. J. M., Groenink, M., & COMPARE study group. (2010). Losartan therapy in adults with Marfan syndrome: Study protocol of the multi-center randomized controlled COMPARE trial. *Trials*, *11*, 3. https://doi.org/10.1186/1745-6215-11-3

Yuan, S.-M., & Jing, H. (2010). Marfan's syndrome: An overview. *Sao Paulo Medical Journal = Revista Paulista De Medicina*, *128*(6), 360–366. https://doi.org/10.1590/s1516-31802010000600009

Robinson, P. N. (2012). Deep phenotyping for precision medicine [_eprint: https://onlinelibrary.wiley.com/doi/p *Human Mutation*, *33*(5), 777–780. https://doi.org/10.1002/humu.22080

Girdea, M., Dumitriu, S., Fiume, M., Bowdin, S., Boycott, K. M., Chénier, S., Chitayat, D., Faghfoury, H., Meyn, M. S., Ray, P. N., So, J., Stavropoulos, D. J., & Brudno, M.

(2013). PhenoTips: Patient phenotyping software for clinical and research use. *Human Mutation*, *34*(8), 1057–1065. https://doi.org/10.1002/humu.22347

Cherif, J., Mjid, M., Ladhar, A., Toujani, S., Mokadem, S., Louzir, B., Mehiri, N., & Béji, M. (2014). Diagnosis delay of pleural and pulmonary tuberculosis. *Revue de pneumologie clinique*, *70*(4), 189–194. https://doi.org/10.1016/j.pneumo.2013.10.005

Kumar, A., & Agarwal, S. (2014). Marfan syndrome: An eyesight of syndrome. *Meta Gene*, *2*, 96–105. https://doi.org/10.1016/j.mgene.2013.10.008

Groza, T., Köhler, S., Moldenhauer, D., Vasilevsky, N., Baynam, G., Zemojtel, T., Schriml, L. M., Kibbe, W. A., Schofield, P. N., Beck, T., Vasant, D., Brookes, A. J., Zankl, A., Washington, N. L., Mungall, C. J., Lewis, S. E., Haendel, M. A., Parkinson, H., & Robinson, P. N. (2015). The Human Phenotype Ontology: Semantic Unification of Common and Rare Disease. *American Journal of Human Genetics*, *97*(1), 111–124. https://doi.org/10.1016/j.ajhg.2015.05.020

König, I. R., Fuchs, O., Hansen, G., von Mutius, E., & Kopp, M. V. (2017). What is precision medicine? *The European Respiratory Journal*, *50*(4), 1700391. https://doi.org/10.1183/13993003.00391-2017

Pavan, S., Rommel, K., Mateo Marquina, M. E., Höhn, S., Lanneau, V., & Rath, A. (2017). Clinical Practice Guidelines for Rare Diseases: The Orphanet Database. *PLoS ONE*, *12*(1), e0170365. https://doi.org/10.1371/journal.pone.0170365

Haendel, M. A., Chute, C. G., & Robinson, P. N. (2018). Classification, Ontology, and Precision Medicine. *The New England Journal of Medicine*, *379*(15), 1452–1462. https://doi.org/10.1056/NEJMra1615014

Amberger, J. S., Bocchini, C. A., Scott, A. F., & Hamosh, A. (2019). OMIM.org: Leveraging knowledge across phenotype–gene relationships. *Nucleic Acids Research*, *47*(Database issue), D1038–D1043. https://doi.org/10.1093/nar/gky1151

Ferreira, C. R. (2019). The burden of rare diseases. *American Journal of Medical Genetics Part A*, *179*(6), 885–892. https://doi.org/10.1002/ajmg.a.61124

Chang, W. H., Mashouri, P., Lozano, A. X., Johnstone, B., Husić, M., Olry, A., Maiella, S., Balci, T. B., Sawyer, S. L., Robinson, P. N., Rath, A., & Brudno, M. (2020). Phenotate: Crowdsourcing phenotype annotations as exercises in undergraduate classes [Number: 8 Publisher: Nature Publishing Group]. *Genetics in Medicine*, *22*(8), 1391–1400. https://doi.org/10.1038/s41436-020-0812-7

Haendel, M., Vasilevsky, N., Unni, D., Bologa, C., Harris, N., Rehm, H., Hamosh, A., Baynam, G., Groza, T., McMurry, J., Dawkins, H., Rath, A., Thaxon, C., Bocci, G., Joachimiak, M. P., Köhler, S., Robinson, P. N., Mungall, C., & Oprea, T. I. (2020). How many rare

diseases are there? *Nature reviews. Drug discovery*, *19*(2), 77–78. https://doi.org/10.1038/d41573-019-00180-y

Harrison, J. E., Weber, S., Jakob, R., & Chute, C. G. (2021). ICD-11: An international classification of diseases for the twenty-first century. *BMC Medical Informatics and Decision Making*, *21*(6), 206. https://doi.org/10.1186/s12911-021-01534-6

Jacobsen, J., Baudis, M., Baynam, G., Beckmann, J., Beltran, S., Callahan, T., Chute, C., Courtot, M., Danis, D., Elemento, O., Freimuth, R., Gargano, M., Groza, T., Hamosh, A., Harris, N., Kaliyaperumal, R., Khalifa, A., Krawitz, P., Koehler, S., & Robinson, P. (2021). *The GA4GH Phenopacket schema: A computable representation of clinical data for precision medicine*. https://doi.org/10.1101/2021.11.27.21266944

Köhler, S., Gargano, M., Matentzoglu, N., Carmody, L. C., Lewis-Smith, D., Vasilevsky, N. A., Danis, D., Balagura, G., Baynam, G., Brower, A. M., Callahan, T. J., Chute, C. G., Est, J. L., Galer, P. D., Ganesan, S., Griese, M., Haimel, M., Pazmandi, J., Hanauer, M., … Robinson, P. N. (2021). The Human Phenotype Ontology in 2021. *Nucleic Acids Research*, *49*(D1), D1207–D1217. https://doi.org/10.1093/nar/gkaa1043

Nayak, S. S., Schneeberger, P. E., Patil, S. J., Arun, K. M., Suresh, P. V., Kiran, V. S., Siddaiah, S., Maiya, S., Venkatachalagupta, S. K., Kausthubham, N., Kortüm, F., Rau, I., Wey-Fabrizius, A., Van Den Heuvel, L., Meester, J., Van Laer, L., Shukla, A., Loeys, B., Girisha, K. M., & Kutsche, K. (2021). Clinically relevant variants in a large cohort of Indian patients with Marfan syndrome and related disorders identified by next-generation sequencing. *Scientific Reports*, *11*(1), 764. https://doi.org/10.1038/s41598-020-80755-7

Tehrani, A. Y., White, Z., Milad, N., Esfandiarei, M., Seidman, M. A., & Bernatchez, P. (2021). Blood pressure□independent inhibition of Marfan aortic root widening by the angiotensin II receptor blocker valsartan. *Physiological Reports*, *9*(10), e14877. https://doi.org/10.14814/phy2.14877

Steinhaus, R., Proft, S., Seelow, E., Schalau, T., Robinson, P. N., & Seelow, D. (2022). Deep phenotyping: Symptom annotation made simple with SAMS. *Nucleic Acids Research*, gkac329. https://doi.org/10.1093/nar/gkac329

Acronyms

BfArM Federal Institute for Drugs and Medical Devices. 5

BIH Berlin Institute of Health at Charité. 4

CSS Cascading Style Sheets. 15

DAG Directed Acyclic Graph. 6

DIMDI Deutsches Institut für Medizinische Dokumentation und Information. 5, 14

FK Foreign Key. 16, 27

GA4GH Global Alliance For Genomics and Health. 8, 13, 14, 32, 38, 44, 46, 56

HPO Human Phenotype Ontology. 4, 6, 7, 9, 14, 17, 24

HTML Hypertext Markup Language. 15, 51

ICD International Statistical Classification of Diseases and Related Health Problems. 4, 5

JSON JavaScript Object Notation. 9, 21, 27, 34

MFS Marfan syndrome. 3

MONDO Mondo Disease Ontology. 9, 38

OMIM Online Mendelian Inheritance in Man. 4, 6, 14, 17, 24

ORDO Orphanet Rare Disease Ontology. 9

Orphanet Orphanet Database. 4, 6, 7, 14, 17, 24

PK Primary Key. 16

PM Precision Medicine. 1

RD Rare Disease. 1, 3, 7

SAMS Symptom Annotation Made Simple. 4, 5, 13, 14, 19, 32, 51, 55

UI User Interface. 15

UTC Coordinated Universal Time. 17

YAML YAML Ain't Markup Language. 9, 34

Glossary

deep phenotyping Is the precise and comprehensive analysis of phenotypes and their phenotypic abnormalities. 1

phenotype Is the combined phenotypic abnormalitiss. 1

precision medicine Is the stratification of patients to provide the best medical care. 1