Interface for description and analyses of systemic oncology protocols

Georgina Kennedy[1,2,3], Travis Zack[4], Ivy Cerelia Valerie[1], Michael Gurley[5], Jeremy Warner[6]

[1] Faculty of Medicine & Health, UNSW Sydney, [2] Ingham Institute of Applied Medical Research, Liverpool, Sydney, [3] Maridulu Budyari Gumal (SPHERE) Cancer Clinical Academic Group, Australia, [4] University of California, San Francisco, CA, USA, [5] Northwestern University, Chicago, IL, United States, [6] Brown University and Lifespan Cancer Institute, Providence, RI, USA.

Background

Medical oncology is a complex and rapidly advancing field, and the number of utilized systemic therapies in the real world continues to expand [1]. This proliferation of distinct, yet similar, systemic therapy regimens, combined with their frequent modifications in real-world settings, necessitates detailed and systematic baseline representations to support scalable observational research. A comprehensive representation of a systemic therapy regimen includes information on approved patient populations, treatment components, dosages, periodicity, and duration. HemOnc, a leading tool curated by oncology specialists, serves as a comprehensive resource for cataloguing such data in approved regimens for systemic therapy in medical oncology [2,3]. In part due to this comprehensive nature and highly detailed specification, the learning curve for directly handling and utilizing this data effectively can be challenging for those unfamiliar with the nuances of regimen definitions and interrelationships. To flatten this familiarization curve, as well as enhance the utility of this model to the growing community of real-world-evidence researchers with Python expertise, we introduce a new interface designed to facilitate the intuitive extraction and summarization of detailed protocol schedule information, supporting a variety of downstream tasks in medical oncology research. This interface is an extension module to an existing object relational model that has been defined for handling of the OMOP Common Data model (CDM) [4,5], ensuring seamless integration with underlying concepts and exposures, and enabling the direct application of these baseline regimen definitions to OMOP data sources.

Methods

Our source data is the HemOnc ontology, which is composed of controlled terms and relationship metadata intended to fully describe the evolving practice of anti-cancer treatments and the relevant supporting evidence. Inputs to this database are limited to oncology regimens that are backed by at least phase II clinical data and have been published in peer-reviewed medical journals. Each entry includes the complete description of how the regimen was specified in the relevant clinical trial, and thus provides the details of how it is expected to be administered in the clinical setting. This necessitates inclusion of not only pharmaceutical agents and their links to base regimen, therapeutic contexts, and source evidence, but also the specification of many complex relative and absolute scheduling factors. These schedules frequently include irregular and/or multipart *signature (sig)* specifications at both the cycle and component level, often having extremely subtle differences between regimen variants.

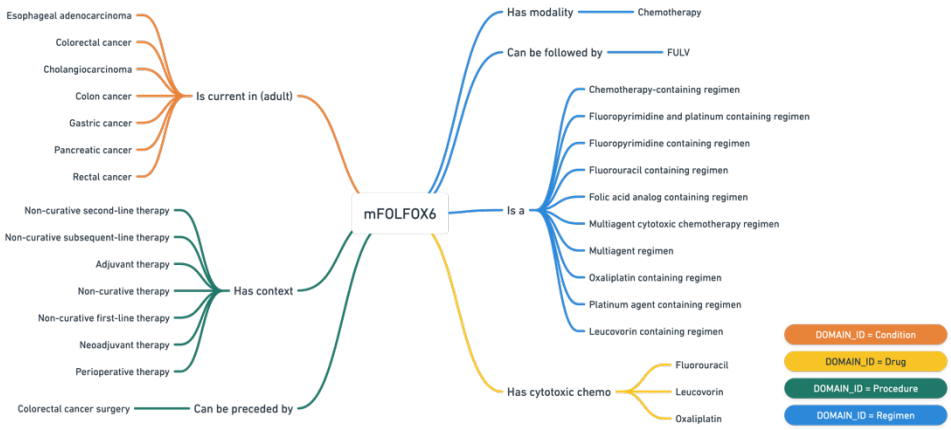


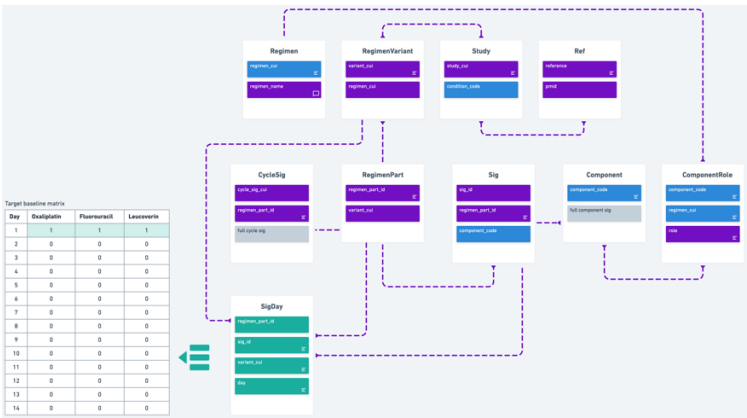
Figure 1: Example existing HemOnc OMOP relationships showing selected relationships centred around the mFOLFOX6 regimen in condition, drug, procedure and regimen domains.

Although the regimen properties or metadata such as inheritance / classification are easily expressed in the typical OMOP triples (e.g. figure 1), there are many richer elements of the full sig specifications that are challenging to fit within this form, such as in the example presented in figure 2. In order to structure both the component-level and cycle-level sigs, the breadth of variables becomes unwieldy, and we have therefore chosen instead to retain the wide format of the HemOnc Sig table specification while retaining tight integration with the OMOP vocabulary and clinical tables.

Figure 2: Example modelled schedule specification by relationship triples (multipart regimen containing only regular cycle and component sigs)

Results

We ingested the source HemOnc files into a relational database (in this case, sqlite for maximal accessibility and convenient sharing, however this is not prescriptive), with the relationships defined according to a SQLAchemy object-relational model (ORM) [6, 7]. Within this ORM, we have further extended the existing HemOnc *Sig* definitions to support convenient direct application of regimen schedule definitions as baselines when inferring intended treatment as well as making comparisons between this intended treatment and actual delivered dosage (Figure 3).



component_name	step_number	route	doseMinNum	doseMaxNum	component_class	sig_id	day
Cyclophosphamide	1 of 1	IV	500	500	IV intermittent canonical Sig	0	1
Epirubicin	1 of 1	IV	100	100	IV intermittent canonical Sig	1	1
Fluorouracil	1 of 1	IV	500	500	IV intermittent canonical Sig	2	1

Figure 3: Illustrative subset of HemOnc relationships to define interface – blue fields integrate directly (via concept_code) to OMOP vocabularies. SigDay extension provided to support production of target base cycle matrices.

It should be noted that this extension to the model does not add any new information that was not already present and acts simply as a convenience measure for a common use-case that has been very challenging to date. This format enhances the usability of the HemOnc model through the documentation and enforcement of relationships in a way that is familiar to many researchers and analysts.

To demonstrate the utility of this interface, we have produced some descriptive analyses that show the changing nature of chemotherapy regimens over time (Figure 4).

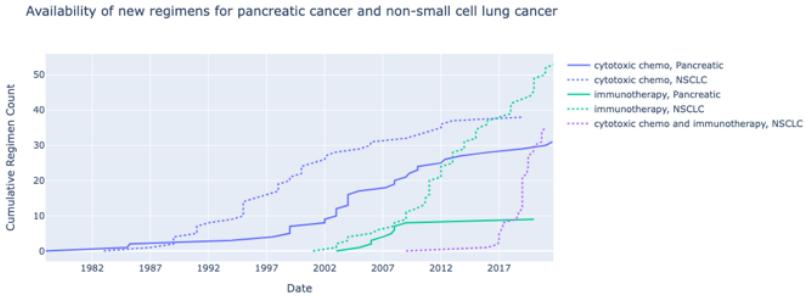


Figure 4: Example descriptive analyses created with the HemOnc Alchemy interface

Conclusion

The rapid rate of change of anticancer treatment practice necessitates a highly detailed structured representation of the full regimen specification. Many of these regimen properties can be expressed in the OMOP CDM paradigm, however their complete form requires some additional data that are not a good fit for this form. We have demonstrated an interface that can integrate neatly with existing CDM data and tools in a way that makes this full richness available at scale to a new audience of researchers and developers, without affecting the core CDM definition itself.

References

1. Scott, E. C., Baines, A. C., Gong, Y., Moore Jr, R., Pamuk, G. E., Saber, H., ... & Beaver, J. A. (2023). Trends in the approval of cancer therapies by the FDA in the twenty-first century. Nature Reviews Drug Discovery, 22(8), 625-640.
2. Warner, J. L., Dymshyts, D., Reich, C. G., Gurley, M. J., Hochheiser, H., Moldwin, Z. H., ... & Yang, P. C. (2019). HemOnc: A new standard vocabulary for chemotherapy regimen representation in the OMOP common data model. Journal of biomedical informatics, 96, 103239.
3. Travis, Z. A. C. K., & Warner, J. L. (2024). Introducing a Comprehensive Score of Systemic Anticancer Treatment Relevance. Studies in health technology and informatics, 310, 464.
4. https://github.com/AustralianCancerDataNetwork/OMOP_Alchemy
5. Kennedy, G. & Churches, T., Using Object Relational Mapping to Improve Sustainability of a Research-ready Clinical Cancer Data Platform, OHDSI APAC Symposium, 2023
6. https://github.com/AustralianCancerDataNetwork/HemOnc_Alchemy
7. https://www.sqlalchemy.org/

Supplementary Material

S1. Entity Relationship Diagram

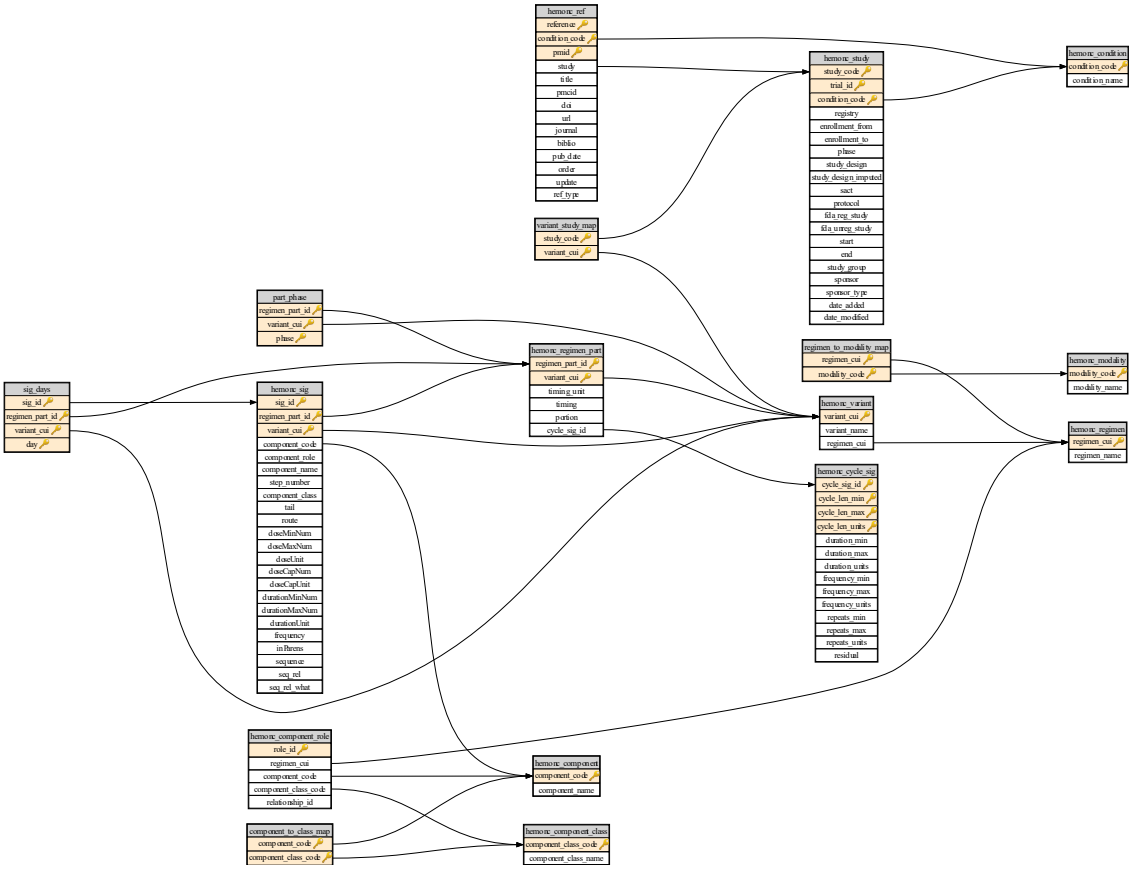


Figure 5: Entity relationship diagram auto-generated from the complete interface definition using the fastlite library (available at https://github.com/AnswerDotAI/fastlite)

S2. Query 1 (cf. figure 3):

SQLAlchemy query to select SigDay matrix as per figure 3

```
var_component_days = session.query(Hemonc_Regimen.regimen_name,
                                   Hemonc_Regimen.regimen_cui,
                                   Hemonc_Variant.variant_name,
                                   Hemonc_Variant.variant_cui,
                                   Hemonc_Cycle_Sig.cycle_len_min,
                                   Hemonc_Cycle_Sig.cycle_len_max,
                                   Hemonc_Cycle_Sig.cycle_len_units,
                                   Hemonc_Cycle_Sig.frequency_min,
                                   Hemonc_Cycle_Sig.frequency_max,
                                   Hemonc_Cycle_Sig.frequency_units,
                                   Hemonc_Cycle_Sig.residual,
                                   Hemonc_Regimen_Part.regimen_part_id,
                                   Hemonc_Regimen_Part.timing,
                                   Hemonc_Regimen_Part.timing_unit,
                                   Hemonc_Regimen_Part.portion,
                                   Hemonc_Sig.frequency,
                                   Hemonc_Sig.component_name,
                                   Hemonc_Sig.component_role,
                                   Hemonc_Sig.step_number,
                                   Hemonc_Sig.route,
                                   Hemonc_Sig.doseMinNum,
                                   Hemonc_Sig.doseMaxNum,
                                   Hemonc_Sig.component_class,
                                   Hemonc_Sig.tail,
                                   Sig_Days.sig_id,
                                   Sig_Days.day
                                  ).join(Hemonc_Variant, Hemonc_Variant.regimen_cui == Hemonc_Regimen.regimen_cui
                                  ).join(Hemonc_Regimen_Part, Hemonc_Regimen_Part.variant_cui==Hemonc_Variant.variant_cui, isouter=True
                                  ).join(Hemonc_Cycle_Sig, Hemonc_Cycle_Sig.cycle_sig_id==Hemonc_Regimen_Part.cycle_sig_id, isouter=True
                                  ).join(Hemonc_Sig, sa.and_(Hemonc_Sig.variant_cui==Hemonc_Regimen_Part.variant_cui,
                                                             Hemonc_Sig.regimen_part_id==Hemonc_Regimen_Part.regimen_part_id),
                                                             isouter=True
                                  ).join(Sig_Days, sa.and_(Sig_Days.variant_cui==Hemonc_Sig.variant_cui,
                                                           Sig_Days.regimen_part_id==Hemonc_Sig.regimen_part_id,
                                                           Sig_Days.sig_id==Hemonc_Sig.sig_id)).all())
```

Equivalent compiled SQL autogenerated from SQLAlchemy query above:

```
SELECT hemonc_regimen.regimen_name,
       hemonc_regimen.regimen_cui,
       hemonc_variant.variant_name,
       hemonc_variant.variant_cui,
       hemonc_cycle_sig.cycle_len_min,
       hemonc_cycle_sig.cycle_len_max,
       hemonc_cycle_sig.cycle_len_units,
       hemonc_cycle_sig.frequency_min,
       hemonc_cycle_sig.frequency_max,
       hemonc_cycle_sig.frequency_units,
       hemonc_cycle_sig.residual,
       hemonc_regimen_part.regimen_part_id,
       hemonc_regimen_part.timing,
       hemonc_regimen_part.timing_unit,
       hemonc_regimen_part.portion,
       hemonc_sig.frequency,
       hemonc_sig.component_name,
       hemonc_sig.component_role,
       hemonc_sig.step_number,
       hemonc_sig.route,
       hemonc_sig."doseminnum",
       hemonc_sig."dosemaxnum",
       hemonc_sig.component_class,
       hemonc_sig.tail,
       sig_days.sig_id,
       sig_days.day
FROM   hemonc_regimen
       JOIN hemonc_variant
         ON hemonc_variant.regimen_cui = hemonc_regimen.regimen_cui
       LEFT OUTER JOIN hemonc_regimen_part
                    ON hemonc_regimen_part.variant_cui =
                       hemonc_variant.variant_cui
       LEFT OUTER JOIN hemonc_cycle_sig
                    ON hemonc_cycle_sig.cycle_sig_id =
                       hemonc_regimen_part.cycle_sig_id
       LEFT OUTER JOIN hemonc_sig
                    ON hemonc_sig.variant_cui = hemonc_regimen_part.variant_cui
                       AND hemonc_sig.regimen_part_id =
                           hemonc_regimen_part.regimen_part_id
       JOIN sig_days
         ON sig_days.variant_cui = hemonc_sig.variant_cui
            AND sig_days.regimen_part_id = hemonc_sig.regimen_part_id
            AND sig_days.sig_id = hemonc_sig.sig_id
```

Example converted Sig matrix from above query to use in regimen detection algorithm

component_name	Cyclophosphamide	Epirubicin	Fluorouracil
0	1.0	1.0	1.0
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.0	0.0	0.0
4	0.0	0.0	0.0
5	0.0	0.0	0.0
6	0.0	0.0	0.0
7	0.0	0.0	0.0
8	0.0	0.0	0.0
9	0.0	0.0	0.0
10	0.0	0.0	0.0
11	0.0	0.0	0.0
12	0.0	0.0	0.0
13	0.0	0.0	0.0
14	0.0	0.0	0.0
15	0.0	0.0	0.0
16	0.0	0.0	0.0
17	0.0	0.0	0.0
18	0.0	0.0	0.0
19	0.0	0.0	0.0
20	0.0	0.0	0.0

S3. Query 2 (cf. figure 4):

SQLAlchemy query to track availability of new regimens, variants and evidence over time, per figure 4:

```
q = session.query(Hemonc_Regimen.regimen_cui,
                  Hemonc_Regimen.regimen_name,
                  Hemonc_Variant.variant_name,
                  Hemonc_Variant.variant_cui,
                  Hemonc_Study.study_code,
                  Hemonc_Study.start,
                  Hemonc_Study.end,
                  Hemonc_Study.sponsor_type,
                  Hemonc_Study.enrollment_from,
                  Hemonc_Study.enrollment_to,
                  Hemonc_Ref.title,
                  Hemonc_Ref.pub_date,
                  Hemonc_Condition.condition_name
                  ).join(Hemonc_Variant,
                         Hemonc_Variant.regimen_cui == Hemonc_Regimen.regimen_cui, isouter=True
                  ).join(variant_study_map,
                         variant_study_map.c.variant_cui==Hemonc_Variant.variant_cui, isouter=True
                  ).join(Hemonc_Study,
                         Hemonc_Study.study_code == variant_study_map.c.study_code, isouter=True
                  ).join(Hemonc_Ref,
                         Hemonc_Study.study_code == Hemonc_Ref.study, isouter=True
                  ).join(Hemonc_Condition,
                         Hemonc_Condition.condition_code == Hemonc_Study.condition_code, isouter=True)
```

Equivalent compiled SQL autogenerated from SQLAlchemy query above:

```
SELECT hemonc_regimen.regimen_cui,
       hemonc_regimen.regimen_name,
       hemonc_variant.variant_name,
       hemonc_variant.variant_cui,
       hemonc_study.study_code,
       hemonc_study.start,
       hemonc_study."end",
       hemonc_study.sponsor_type,
       hemonc_study.enrollment_from,
       hemonc_study.enrollment_to,
       hemonc_ref.title,
       hemonc_ref.pub_date,
       hemonc_condition.condition_name
FROM   hemonc_regimen
       LEFT OUTER JOIN hemonc_variant
                    ON hemonc_variant.regimen_cui = hemonc_regimen.regimen_cui
       LEFT OUTER JOIN variant_study_map
                    ON variant_study_map.variant_cui =
                       hemonc_variant.variant_cui
       LEFT OUTER JOIN hemonc_study
                    ON hemonc_study.study_code = variant_study_map.study_code
       LEFT OUTER JOIN hemonc_ref
                    ON hemonc_study.study_code = hemonc_ref.study
       LEFT OUTER JOIN hemonc_condition
                    ON hemonc_condition.condition_code =
                       hemonc_study.condition_code
```