No.	質問	回答
1	ノードとエッジの構造化はマニュアル操作ですか？	基本的にマニュアル操作になります。ある情報をどのように構造化するか（どのように主語-述語-目的語）として記述するか、というのは必ずしも一意に決まるものでもないので、目的に応じて適切に構造化する必要があります。
2	LLMのファインチューニングのコストとRDFのデータ更新のコストの比較データはありますか？	興味深いご質問ですが、私の知る限りではないと思います。
3	1つの概念に対して複数のオントロジーの取り方があると思います。オントロジーの取り方の違いは、とりもなおさず知識構造の違いにつながると思いますが、オントロジーはどのように切り分けているのでしょうか？	ご指摘のように、ある概念（や概念体系）に対して、一般的に複数のオントロジーが存在します。従って、どのように現実の世界をモデル化（構造化）しているのかについては、モデル化する人の考え方や、目的に依存しているとしか言えないと思います。
4	RDFとプロパティーグラフの利点・欠点比較、互換性について教えてください。	RDFの利点は、 ＊ 標準技術があるので、特定の企業に依存しない ＊ URIを使うので、事物をユニークに記述できる（URIは、国際的な標準技術なので） ＊ 異なる組織が独立に作ったRDFを、そのまま統合して使える（フォーマットが標準なので） ＊ OWLなどのオントロジー記述言語も標準技術なので、意味の記述や推論も標準化されている（ただし現実的にはそこまでうまくいっていませんが） RDFの欠点は、 ＊ 利点としていることは、一方で、努力目標のようなところがあり、現実的にはそんなにうまくいかないことも多い 　＊ 例えば、URIはユニークに物事を参照できますが、一方で、ある物事を複数の異なるURIで参照することができるうえ、どのURIを記述の際に使うかはRDF開発者の自由にまかされている、など ＊ 全般的に、開発者向けの技術で、初心者に優しくはないところがある プロパティーグラフの利点は、 ＊ エッジに複数の情報（プロパティ）を持たすことができる（RDFでもできないことはないが、やや複雑な記法になります） ＊ 実装毎に様々な検索ツール可視化ツールが用意されていて、容易に使うことができる プロパティーグラフの欠点は、 ＊ 特定の企業のプロダクトに依存している ＊ 独立して作られた他のプロパティグラフと統合して利用するような用途にはあまり向いていない ＊ 巨大なデータに対してスケールしないケースも多い 両者の互換性については、特にオフィシャルには準備されていないと思いますが、サードパーティのソフトウェアで両者を変換する仕組みなどが提供されています。 例: G2GML https://g2gml.readthedocs.io/en/latest/index.html https://g2gml.readthedocs.io/en/latest/contents/reference.html
5	川島さんにおたずねします。RDF化した情報について、古い情報を新しく更新する場合はどのような処理や管理が必要になるのでしょうか？大量の更新がある場合、仮に新しく情報を追加するだけとなると、そのままトリプルが増えていく、ようなイメージを持ってしまったのですが、それは何かよい管理方法があるのでしょうか？	鋭いご指摘です。RDFのバージョン管理、のようなものは、オフィシャルに定義されていません。 RDFは、W3Cが標準を作っていることからもわかるように、もともとインターネット技術の上に知識ベース的なものを構築しようとする試みです。インターネットのURI（URL）は、そのものは常に現在のものになります。ご指摘のように、新しいトリプルは単純には追加されていく一方になります。 従って、RDFのバージョン管理をするには、なにか取り決めが必要になります。 RDFデータセットにバージョンがある場合（UniProtやPubChemなど）は、そのまま古いRDFはデータベースソフトから消去して、あたらしいRDFデータセットに置き換えるというのが簡単だと思います。 古い情報を維持したまま、新しいものを提供したい場合、例えば、URIに日付を含めるというような工夫が必要になるかもしれません。

No.	質問	回答
6	RDF初心者です。RDFを使えば、述語がわかれば複数のDBを繋いで使える、という理解をしています。TogoIDが必要な背景としては、述語が複雑多様すぎて、適切なSPARQLクエリを構築するのが現実的には難しい、ということがあるのでしょうか？	複雑多様というよりは、厳密に書かれていないケースがよくある、というのがあります。rdfs:seeAlso や oboInOwl:hasDbXref など、何か関係があるという程度の意味でつながれている、というような場合です。TogoID ではもう少し厳密に関係を書いていますので、セマンティックにより正確です。つながれている意味の理解が曖昧でなければそのまま用いて問題ないです。 別のユースケースとして、TogoID では ID 関係の部分だけを切り出していますので、個別にエンドポイントを立てるときに、そこで必要になる turtle ファイルだけをロードすることでコンパクトにまとめることができる、ということが挙げられると思います。
7	池田先生に質問です。Gene symbolが一意でない遺伝子があると理解しておりますが、現実問題としてGene symbol名しか情報として得られていない場合のベストプラクティスなどはありますでしょうか?	「遺伝子名」というのは ”ATP binding cassette subfamily A member 1” などのフルネームのことを指しますので、混同はされていないと思います。Gene symbol は生物種の中では一意のはずですが、シノニムまで含めると一意でなくなります。一般的なベストプラクティスというのは難しいですが、例えばヒトの場合だと遺伝子シンボルを管理している HGNC という機関が、「以前のシンボル」を他のシノニムとは区別して管理しています。いつの時点のシンボル、というのを指定して検索できるかは定かでないですが、参考にできる可能性があります。
8	池田さんに質問です。togoVarのRDFの仕組みの中で、ゲノム座標はどのように取り扱われているのでしょうか。一塩基だけですと、トリプルの中に入れられそうなのですが、構造バリアントなどの情報だと領域のような情報になると思います。たしか、TogoVarでも構造バリアントが登録されていると思いました。 なぜそのような質問をするかというと、ゲノム座標を中心として、情報がつなげられると、解析結果からえられるゲノム座標情報を他の情報につなげられると思っているのですが、そのようなことは可能なのでしょうか?	TogoVar開発者の一人の川島がお答えします。 TogoVar RDF では、FALDOオントロジーでゲノム座標を記述しています。 UniProt やEnsembl, DDBJ などのRDFでも、FALDOを用いて記述しています。 FLADOは、ゲノムの領域に対してアノテーションを行うオントロジーで、例えば以下のような記述になります。 @prefix faldo: <http://biohackathon.org/resource/faldo#> . @prefix so: <http://purl.obolibrary.org/obo/so#> . @prefix vcf: <http://example.org/vcf#> . @prefix ex: <http://example.org/variant/> . ex:var1 a so:sequence_alteration ; rdfs:label ”5-bp insertion” ; vcf:referenceAllele ”ATATG” ; vcf:alternativeAllele ”CTATGAC” ; faldo:location ex:var1_location . ex:var1_location a faldo:Region ; faldo:begin ex:pos_start ; faldo:end ex:pos_end . # 変異開始位置 ex:pos_start a faldo:ExactPosition, faldo:ForwardStrandPosition ; faldo:position 7570000 ; faldo:reference <http://identifiers.org/insdc/NC_000017.11> . # 変異終了位置 ex:pos_end a faldo:ExactPosition, faldo:ForwardStrandPosition ; faldo:position 7570004 ; faldo:reference <http://identifier.org/insdc/NC_000017.11> . 上の例のように、faldo:begin と faldo:end の組み合わせで、ゲノムの特定の領域を表現することができます。ここで、vcf: とか、ex: というのは、例のための適当な語彙になります。 ご指摘のように、ゲノム座標を中心として、情報をつなぐ目的でFALDOが開発され、複数のRDFで使われています。 もう一点、FALDOでは参照しているゲノムの情報は外部に依存しています（faldo:referenceの目的語の部分）。ここが共通化していないと、ご指摘のような情報統合ができません。このためDBCLSでは、Human Chromosome Ontology（HCO）というものも開発しております。

No.	質問	回答
9	NANDOを作る際に、Mondoの拡張ではなく1から作ったのはなぜですか？	NANDOを構築するにあたり、Mondoの拡張という形ではなく、あえて一から作成した理由は、日本の難病制度における疾患分類や病型の定義、制度的な要件が国際的なオントロジー（たとえばMondoやOrphanet）とは異なるためです。 具体的には、Mondoには存在しない「制度に基づく病型（たとえば発症時期や重症度による分類）」が日本の制度には含まれており、これらを忠実に表現するためには、独自の構造でオントロジーを設計する必要がありました。 そのため、国際的な互換性を考慮しつつも、日本固有の制度的背景や運用実態を正確に反映できるよう、NANDOを独自に設計・構築する方針を採りました。
10	MondoとNANDOのマッピングは何で行ったのでしょうか？	MondoとNANDOのマッピングは、まず機械的な候補マッピングを行い、その後に臨床医を含む専門家によるキュレーションを実施するという2段階のプロセスで行いました。 最初の機械的マッピングでは、疾患名に対して正規化処理やステミングを施し、文字列類似度を用いて、Mondoの候補用語を自動的に抽出しました。この段階で、複数の候補語が提示される場合もあります。 次に、提示された候補に対して臨床的意味の一致や制度的背景を考慮しながら、人手で最も適切なMondo用語を選定しました。このキュレーション作業は、疾患の定義・病型分類などを踏まえて慎重に行われています。
11	PubCaseFinderが事前に学習したデータには、症例数の多寡により論文数に差があるため、疾患により偏りが生じてしまうものと推測します。症状の類似度がほぼ同じ疾患が複数ある場合、出力される類似度ランキングは学習したデータ量により影響が出るのでしょうか？	現在のPubCaseFinderでは、疾患ごとの症例数や論文数などの文献情報を学習データとして利用していません。そのため、特定の疾患に関する文献数の多寡が、類似度ランキングに直接影響を与えることはありません。 PubCaseFinderでは、ユーザーが入力したHPO（Human Phenotype Ontology）用語と、それぞれの疾患に登録されているHPO用語の情報量に基づく類似度スコアを計算してランキングを生成しています。この情報量は、文献数ではなくオントロジーの構造から導出されており、階層的に下位に位置する、より具体的なHPO用語ほど情報量が高くなるという特徴があります。 つまり、症状の類似度がほぼ同じ疾患が複数あった場合には、入力された用語の情報量の違い（＝どれだけ詳細で限定的か）や、それらの用語が疾患ごとにどの程度一致しているかがスコアに影響します。学習済みデータ量の偏りによってスコアが変動することはありません。
12	NanbyoDataにある疾患原因遺伝子はどのようなクライテリアで選ばれて表示されているのでしょうか？	NANDOから skos:exactMatch で対応づけたMondoの疾患に対して、MedGen、Orphadata、GenCCの各データベースから疾患原因遺伝子の情報を取得しています。なお、GenCCにおいては関連性の分類が複数ありますが、本システムでは「Definitive（確定的）」と評価された遺伝子のみを対象としています。
13	LLMを用いて実験の文献情報も一緒に読み込むことで精度向上は図るという方向性はありますか？	方向性としては考えられるとは思います。全てのサンプルに対して文献情報を一緒に読み込むと入力が膨らみすぎるので、やるとすれば、同音異義語の判断が付かない場合に文献を参照する、といったやり方が考えられそうです。 ただ当面の方針としては、大量の BioSample レコードをどのように捌くかという部分に力を割こうとしています。既に9割の精度が出ていますので、それをさらに上げるために入力を増やすということをするのはまだ後になるかと思います。