PULMONOLOGY

Modern Identifier Systems: Strategies to Optimize Data Reuse and Interoperability

Kwame Boateng, Kofi Mensah, Emelia Adu-Gyamfi, Kojo Asante, Felicia Owusu*

Department of Materials Science and Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

Abstract

In an era of big data, there is increasing optimism that data mining will yield valuable insights. However, in the life sciences, relevant data is not only "big"; it is also highly decentralized across thousands of online databases. Wringing value from such databases depends on the discipline of data science and on the humble bricks and mortar that make integration possible. Identifiers are a core component of this integration infrastructure; drawing on our experience and on work by other groups, we outline ten lessons we have learned about the identifier qualities and best practices that facilitate large-scale data integration. Specifically, we propose actions that identifier practitioners (providers of online repositories, registries, and knowledgebases) should take in the design, provision and reuse of identifiers; we also outline important considerations for those referencing identifiers in various contexts. While the importance and relevance of each lesson will vary by context, there is a need for increased awareness about how to avoid and manage common identifier problems, especially those related to durability and web-accessibility/resolvability.

Introduction

Life science data is evolving to be ever larger, more distributed, and more natively web-based. However, our collective handling of identifiers has lagged behind these advances. Diverse identifier problems (for instance broken links and '`content drift`' [1]) make it difficult to integrate data, and to subsequently derive new knowledge from it. Optimizing web-based identifiers is harder than it appears; there are a number of approaches that may be used for this purpose, but no single one is perfect: Identifiers are reused in different ways for different reasons, by different consumers. Moreover, digital `entities` (e.g., files), physical `entities` (e.g., biosamples), and descriptive `entities` (e.g., 'mitosis') have different requirements for identifiers [2].

Throughout the life sciences, our handling of identifiers needs improvement. Towards this end, several groups (**Supplemental Text S1**) have been converging on identifier standards that are broadly applicable [3, 4, 5, 6]. Building on these efforts and drawing on our experience, we outline the identifier qualities and best practices we consider particularly important in the context of large-scale data integration. In **Lessons 1-9** (**Table 1**) we propose actions for data providers when designing new identifiers, maintaining existing identifiers, as well as when reusing and referencing identifiers from other datasets. In **Lesson 10**, we conclude with guidance for data integrators and redistributors on how best to reference multiple identifiers from diverse sources. Data providers are urged to take a long-term view of the scope and lifecycle of data and the identifiers that they provide, and to consider using existing identifier platforms and services [7] where appropriate.

Throughout this document, the word "must" is reserved for practices that ensure against identifier collision, ambiguity, or inaccessibility; instances of "must" are also often specific to particular design choices. We use the word "should" to convey that the full implications must be understood and carefully weighed before choosing a different course (eg. consistent with IETF RFC2119 [8]). Terms that appear in `fixed-width font` are defined in the supplemental glossary (**Table S2**).

Many of these recommendations are applicable during the planning and identifier conceptualization phase, i.e. before any identifiers are created. The retrofitting (especially lessons 1, 3, 4 and 7) of existing identifiers can sometimes be too difficult or may even make matters worse: for instance changing existing identifiers introduces identifier mapping issues that offset potential benefits. Each of the lessons is relevant to the basic classes of actions in the identifier ecosystem (design, provision, reuse **Table 1**). These actions in turn are relevant to anyone on the spectrum of seven basic roles ranging from those that publish their own data to

those that provide applications on top of others' data (**Figure 1**). No provider is perfect and no two are alike, hence the objective is to learn from each other's diverse experiences. All of the negative examples herein are anonymized variations of real-world identifiers that we have had to work with. In Table 1, we summarize current and future efforts that could facilitate adoption of the recommendations.

	Identifier actions				
Lesson	**Design & creation**	**Provision & maintenance**	**Reuse & referencing**	**Currently available platforms, standards, and services**	**Current and future efforts that would help lower barriers to adoption**
1. Credit any derived content using its original identifier	direct	indirect	direct		Increased awareness
2. Help local identifiers travel well: document prefix and patterns	direct	direct	direct	Registries, resolvers[a]	Registry coordination[a]
3. Design new identifiers for diverse uses by others	direct	indirect	indirect		Identifier designer/ validator[e]
4. Avoid embedding meaning, or relying on it for uniqueness	direct	direct	direct		Increased awareness
5. Opt for simple, durable web resolution	direct	direct	direct	3rd party resolvers[a]	Increased awareness
6. Implement a version-management policy	direct	direct	indirect		Embeddable citation widget[f]
7. Make URIs clear and findable	direct	direct	indirect		3rd party archiving services[g]
8. Do not reassign or delete identifiers	direct	direct	indirect	Protein Identifier Cross Reference[b]	
9. Document the identifiers you issue and use	direct	direct	direct	HCLS dataset descriptions[c]	Increased awareness, Bioschemas.org[h]
10. Reference and display responsibly	direct	direct	direct	Journal Article Tag Suite[d]	Increased awareness[d], Integration of data IDs in Ref managers

Table 1. A summary of the 10 recommendations, their relevance to different kinds of identifier actions, current and future efforts that would help lower barriers to best practice.

a. Registries, 3rd party resolvers: A list of identifier resolvers and identifier registries is in **Supplemental Table S3**.
b. PICR: Protein Identifier Cross-Reference Service [9] has a service that returns identifier mappings, optionally including deleted ones. PICR or a similar service could be developed to have broader scope.
c. HCLS: Health Care and Life Sciences dataset descriptions [10] provide a standard representation of the original sources of data (and therefore identifiers) in any integrated dataset.
d. In the context of the literature, Journal Article Tag Suite [11] provides a standard way for data citations to be represented in the literature, facilitating credit and reward mechanisms. However, outside of the literature, referencing and display is primarily an issue of increasing awareness.
e. Identifier designers could help data producers choose the design that best suits their particular use case, validators could determine whether an existing identifier is valid according to a published scheme.
f. Embeddable citation widgets could help providers display citation information, clearly and consistently.
g. Archiving services: client-facing services include Memento web protocol [12]. We authors of this paper are not aware of any existing platforms that *providers* can outsource their archiving to, but such a service may be worthwhile.

h. BioSchemas.org is promoting more consistent adoption of schema.org markup in the life sciences. [13] Markup can facilitate more transparent provenance and credit mechanisms of integrated data, as well as optimizing data for discovery by search engines, whether Google, or others.

1. Credit any derived content using its original identifier

If you manage an online `database` (`repository`, `registry`, or `knowledgebase`), consider its role in identifying and referencing the knowledge that it publishes. We advise that you only create your own identifiers for new knowledge (**Figure 1**). Wherever you are referring to existing knowledge, do so using existing identifiers (lesson 10): otherwise, wherever the 1:1 relationship of identifier:entity breaks down, costly mapping problems arise. Whether or not you create a new ID, it is vital to credit any derived content using its `indigenous identifiers` [14]; to facilitate data integration, all such identifiers should be machine processable.

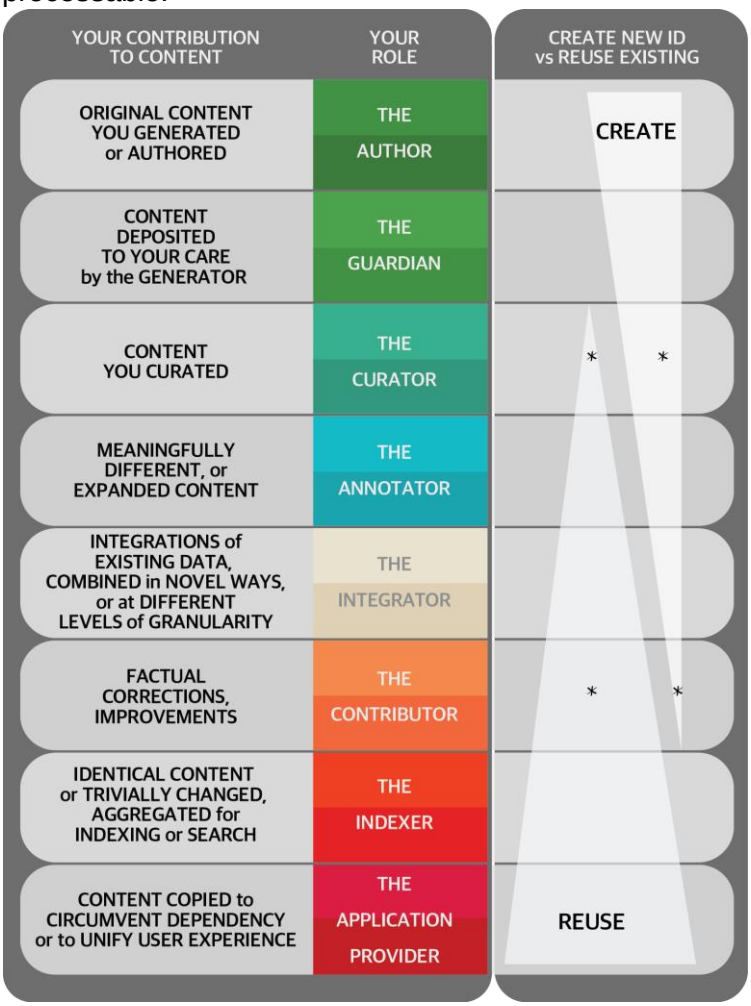

Figure 1. Contributions and roles related to content as they correspond to identifier creation vs reuse.
The decision about whether to create a new identifier, or reuse an existing one depends on the role you play in the creation, editing, and republishing of content; for certain roles (and when several roles apply) that decision is a judgement call. However, if the `indigenous` identifier is not explicitly reused and instead a new identifier is created, we strongly recommend that the `indigenous` identifier be referenced and transparently mapped to the new identifier. In the roles of contributor and/or curator, the best course of action is often to correct/improve the original record in collaboration with the original source; the guidance about ID creation versus reuse is meant to apply only when such collaboration is not practicable (and an alternate record is created).

2. Help local identifiers travel well: document prefix and patterns

Data does not thrive in silos: it is most useful when reused, broken into parts and integrated with other data, for instance in database cross references ("`db xrefs`"). In spite of how important identifiers are to this process, the confusion with identifiers often starts with the basics, including what the "identifier" even is. A `local ID` (**Box 1**) is an identifier guaranteed only to be unique in a given local context (eg. a single provider, a single collection, etc), and sometimes only within a specific version; as such, it is poorly suited to facilitate data integration because it can collide when considered in a more global landscape of many such identifiers. For instance, the `local ID` "9606" corresponds to numerous entities whose local accessions are based on simple digits, including: a Pubmed article, a CGNC gene, a PubChem chemical, as well as an NCBI taxon, a BOLD taxon, and a GRIN taxon. `Local IDs` therefore need to be contextualized in order to be understood and accessed (resolved) on the web. This is often accomplished through the use of a `prefix`, which should be documented. If this is

overwhelming, don't forget that there are third party resolvers and services built to help for exactly this reason (see **Lesson 5**).

Tim Berners-Lee said "cool URIs don't change" [15] because when URIs *do change* (or disappear) all existing references break. In the context of academia alone, "reference rot" problem impacts one in five publications [16]. Despite link rot vulnerability, the global http/s URI (**Box 1**) is the best available identifier form for machine-driven global data integration because the http URI is a) a widely adopted IETF standard and b) its uniqueness is ensured by a single well-established name-granting process (DNS). However, the length of URIs can make them unwieldy for tasks involving human readability, even within structured machine-parsable documents [17]. Compact URIs (CURIEs[18], **Box 1**) are a mature W3C standard that is well established in some contexts (e.g. JSON-LD and RDFa) as they enable URIs to be understood and conveniently expressed. CURIEs are not appropriate for every context (**see Lesson 10**), but they complement http URIs in important ways for data integrators, especially those that re-publish the data they integrate. For instance, the location independence of CURIEs provides certain advantages: first, it straightforward to toggle between an original external http URI and the corresponding record in the integrator's database, depending on what a given application page/stack calls for. Secondly, it is rare that only a single http URI exists for an entity; when several source databases differ on which http URI to reference, CURIEs can provide clues that facilitate careful collapsing of equivalent but differently-represented http URIs. Thus if you are a database provider, it is in your best interests to document and preferably register a) the prefix (**Box 1**) that you would like others to use and b) its binding to a uri pattern (**Box 1**). Your chosen prefix should be unique, at least among datasets that are likely to be used in the same context. **Table S3** contains a list of registries that may be suitable depending on the kind of data. PrefixCommons [19] is a platform designed to enable such registries to be better harmonized and utilized and for any given integrator to publish the mappings that they happen to use.

Box 1. Local and Global Identifier Terminology

An **identifier** is a sequence of characters that identifies an entity.

1. **Local ID** is an identifier that is unique within the scope of a single database.
   - Databases and library systems often refer to the **Local ID** as an 'Accession Number'.
   - **Local ID** formats vary by provider and may have subparts such as entity type (**see Lesson 3)**. A Local ID may be opaque (e.g. A0A022YWF9) or recognizable (e.g. ZDB-GENE-980526-388). It may include an embedded prefix (**ZDB**-GENE-980526-388), a colon-delimited prefix (MGI:80863), or no prefix at all (9606).
2. **Global identifier** is an **identifier** that is guaranteed to be globally unique
   - **Uniform Resource Identifier (URI)** is an identifier that is uniform; it is an ASCII string that uniquely identifies an individual web resource[20]. For simplicity in this paper, by URI we mean only those global URIs a) of type HTTP, HTTPS, etc. b) that are designed to persistently resolve to (provide or redirect to) a webpage containing information about the identified entity. An example of a URI is http://purl.uniprot.org/uniprot/A0A022YWF9.
   - When referring to **compact URIs (CURIES)**, we mean an identifier comprised of **<Prefix>:<Local ID>** wherein **prefix** is deterministically expandable to **a URI pattern** (see below) which *alone* is the basis for the CURIE's global uniqueness. An example of a CURIE is UniProtKB:A0A022YWF9
3. A **URI pattern** is a fixed sequence of characters that can be used to resolve a database's local IDs. In this paper, we mean "URI pattern" to mean the simplest scenario wherein the pattern can be prepended to the local ID (or to the part of the CURIE that follows the colon, if different). In the ZFIN example above, the **prefix** is ZFIN, the **URI pattern** is http://zfin.org/, and the CURIE would be ZFIN:ZDB-GENE-980526-388. When expanded to a full URI, the result would be: http://zfin.org/ZDB-GENE-980526-388. URI patterns vary considerably; see also **Fig 2** and glossary (**Table S2**) for additional terms and concepts.

Table 2. Desirable characteristics for database identifiers in the life sciences

Characteristics	Definition	Rationale/impact on data integration
Unambiguous	One **`Local ID`** must be associated to no more than one entity *locally.* One `URI` must be associated to no more than one entity *globally*	Avoids collisions that result in integrating on the wrong entity
Unique	One entity should ideally be identified by no more than one URI	1) Eliminates the cost of maintaining public mappings between equivalent identifiers 2) Avoids false negatives if data integrators do not leverage or know about a mapping
Stable (identifier)	The `URI, and by extension the local ID` should, wherever possible stay the same over time	Avoids `link rot`
Stable (entity)	Identifier must NOT be reassigned to an altogether different entity, though the original entity may evolve provided a change history is documented	Avoids integrating on the wrong entity
Version-documented	If the entity's definition or essential metadata changes substantially, (Lesson 7) the `identifier` should, wherever posslble be versioned and/or change history documented	Avoids integrating on the wrong entity state (specified through version)
Persistent	The identifier must NOT be deleted (but may be deprecated)	Avoids `link rot`
Web-resolvable	The `URI` must be resolvable to a web address where the data or information about the entry can be accessed	Avoids the unnecessary proliferation of resolvable identifiers issued by third parties (for entities that are not resolvable and/or not identified in their native context) See also `surrogate identifier.`
Convertible	The **`Local ID`** and its `URI` counterpart must be inter-convertible by applying the **`URI pattern`** to the **`Local ID.`** Note that in some communities (eg. ontologies), the Local ID is often a `CURIE` by default.	Avoids the need for special handling of edge cases when integrating data at scale
Defined	The total set of assignable identifiers for the `database` must be describable through a formal pattern (regular expression)	Facilitates validation and extraction from scientific text, thus the pattern should be as tightly specified as possible (see lesson 3)
Web-friendly	The **`Local ID`** should wherever possible be of a format that does not need special handling when used in `URLs` and common exchange formats (e.g. XML)	Avoids potential points of failure due to malformed URL, XML, etc.
Free to assign	The `identifier` should ideally be assigned at no cost to individuals depositing data in a repository	Lowers barriers for data generators to deposit data
Open access and use	The `identifier` and its `label` should be able to be transparently referenced and actioned (e.g. in a public index or search) anywhere by anyone and for any reason. Restrictions on associated data may apply but are not recommended.	Enables integration on the basis of scientific merit, rather than on the restrictions of the license
Documented	The identifier scheme should be documented	Encourages consistent use of existing identifiers by others and reduces the number of ways identifiers are represented.

3. Design new identifiers for diverse uses by others

Pre-existing identifiers should be referenced without modifications (see **Lesson 10**). However, when new local identifiers are necessary, there are some design decisions that can facilitate their use in diverse contexts (spreadsheets, other databases, web applications, publications, etc.).

`Local Identifiers`:

1. Should, wherever possible, comprise only letters, numbers and URL-safe delimiters. Omission of other special characters guards against corruption and mistranscription in many contexts; however, it is acceptable that the `Local ID` be in `CURIE` format since modern browsers resolve colons without having to encode them. Although characters "/" and "?" are technically URL-safe, they are very problematic when used *within* the `local ID` as these characters are assumed to have special meaning and can complicate parsing of the identifiers, whatever forms they take.
2. Consider using both letters and numbers. This avoids misinterpretation as numeric data (e.g. truncation of leading zeros or conversion to exponents in spreadsheets).
3. Should avoid patterns that could result in misinterpretation/corruption whether as dates (e.g. "may-15"), exponents (e.g. "5e1234")[21], or as unintended words (e.g. "bad-12"). Such issues in gene names alone have been shown to impact 19% of life sciences papers. [22]
4. Must adhere to a formal pattern (regular expression); this facilitates the validation of URIs and improves the accuracy of mining identifiers from scientific text. Consider a fixed length of 8-16 characters (according to the anticipated number of required **`Local ID`**s). A pattern may be extended if all available identifiers are issued, but existing identifiers should not be changed. To minimize **`Local ID`** collisions at global scale, it is considerate to tightly specify your pattern (e.g. using one or more fixed letters).
5. The regular expression should include a fixed, documented case convention. In most cases, it is advised that identifiers not rely on case for their uniqueness: if you assign `ab-12345` to one entity and AB-12345 to *a different* entity, collisions due to mistranscription are more likely. Case-sensitive patterns are best reserved for when brevity is a constraint (e.g. millions of IDs are required and each ID has to be short enough to be printed on a vial label).
6. Should ideally not contain '.' except to denote version where appropriate (see **Lesson 7**).

Others will reference your identifiers in compact notation, whether in websites, or in structured documents like `JSON-LD`. One minor consideration will make your identifiers better suited to this:

7. A historically common, if thorny, identifier pattern is that '_' and ':' are often interconverted and it has come to be understood as compact notation, delimiting the `prefix` from the rest of the identifier. Therefore '_' or ':' should occur no more than once per identifier and should only be used if local identifiers are intended to be deterministically expanded to a resolvable http URI. For instance, if your intended `prefix` is '`MyDB`', then either `MyDB:gene-6622` or `MyDB_gene-6622` are acceptable patterns, but `MyDB_gene_6622` is problematic as it could result in three possible conversions by others, even if these are not intended: `MyDB_gene:6622`, `MyDB:gene_6622`, `MyDB:gene:6622`. Whatever pattern you adopt, document which variations you support resolution of, if any.

4. Avoid embedding meaning, or relying on it for uniqueness

The structure and scope of collections evolve, as does scientific understanding; minimizing the meaning embedded in identifiers makes them less vulnerable to obsoletion. In human genetics many genes were initially identified based on disease association; later the identification, nomenclature, and function of genes were separated into different activities. Meaning should only be embedded if it is indisputable, unchangeable and also useful to the data consumer (e.g. computer-processable). For instance, the type of entity imparts meaning to users and may fulfill these three criteria. When encountered, typing may be embedded, either within the local ID (`ENSMUSG…`), or within the http URI path (`.../gene/12345`), or both. In any case, if you opt to include type in the identifiers you issue, avoid relying on type for uniqueness: that is to say once a local identifier eg. `12345` is assigned, it should never be recycled for another entity, even an entity of a different type for instance `.../gene/12345` and `.../patient/12345`.

If you need the ability to convey meaning in a dense character space, you don't need to do so in the identifier itself; consider instead implementing an entity `label`, for instance as is done in model organism nomenclature (`label: Kit`^{W}`/Kit`$^{W\text{-}v}$`, id: MGI:2171276). Labels` are for human readability only; even if they are deemed durable, labels should not be treated as identifiers, nor should they appear within HTTP URIs. URI patterns, if type-specific, require a corresponding type-specific `prefix` (e.g. LINCS-cell corresponds to `http://lincs.hms.harvard.edu/db/cells/$id/` whereas LINCS-protein corresponds to `http://lincs.hms.harvard.edu/db/proteins/$id/`). MGI implements both type-agnostic resolution (`http://www.informatics.jax.org/accession/MGI:2442292`) and type-specific destinations (`http://www.informatics.jax.org/marker/MGI:2442292`). Dual approaches like theirs can be helpful to different kinds of consumers: type-agnostic resolution is useful in cases such as data citation in the literature where a) the type of the identified entity is not of primary importance, or b) the type of the entity is already conveyed contextually, and/or c) where resolution is done systematically at scale and/or involves many and varied or volunteer contributors that may be difficult to coordinate. Type-specific resolution is useful in cases like bioinformatic research pipelines where embedded type may facilitate the human-led debugging process. If you support both kinds of resolution, it is best to document whether you intend for both to be treated as persistent.

Whether or not your URIs or your `Local IDs` include type, you should provide other ways for humans and machines to determine the type of entity that is being identified; this is most often achieved via webservices (eg. as done via Monarch API), but ideally also within metadata landing pages [7][23] if provided.

5. Opt for simple, durable web resolution

If you are a database provider, you must implement an `HTTP URI pattern` (**Fig. 1 panel B**) for local identifiers to be "resolvable" to a web page. If you choose to outsource to a resolver service, use an approach that adheres to best practice [7] (e.g. DOI (DataCite, CrossRef), Identifiers.org, Handle.net, PURL (now via InternetArchive), EPIC, ARK) and be mindful of your constraints regarding cost, metadata ownership, turnaround time, etc. (See **Text S4** for a more comprehensive list of considerations.) Some of these resolver services can even provide `content negotiation` for different encodings of your data [7] and make it easier to provide direct access to data, metadata, and persistence statements [23]. If you have the resources to support your own persistent URIs, design these to be "cool"[15]; this is most easily achieved by keeping them simple. Omit anything that is likely to change or lapse, including administrative details (e.g. grant name) or implementation details such as file extensions ('`resource.html`'), query strings ('`param=value`'), and technology choices ('`.php`'). Never embed the local-id in the query part of a URI eg. `http://example.com/explore?record=A123456`. Make every attempt to limit the degree of path nestedness (eg. do `http://example.com/A123456` rather than `http://example.com/vertebrates/mammals/rodents/rat/white-rat/A123456`); see also lesson 4 above regarding types. The compact URI approach can work with any resolver(s): see for instance examples 4 and 5 in **Figure 2**. By choosing a single `URI pattern`, you make it possible for others to resolve your identifiers simply (**Fig. 2 panel A**). In all cases, the `URI pattern` must include the protocol (e.g. http://) and, if applicable, trailing slash or other delimiters. Trailing characters are discouraged as they unnecessarily increase the variability with which the identifier is represented and also complicate straightforward appending of the local ID (requiring that tokens such as `$id` hold the place of the local ID in the URI pattern eg `http://example.com/$id/view.do`). Despite their differences, the examples in Fig. 2 share the most important features; each: 1) has a simple, durable mechanism for resolution, 2) has an HTTP URI that includes the local ID with no modification, 3) omits volatile meaning (or all meaning) from the local ID and from the HTTP URI.

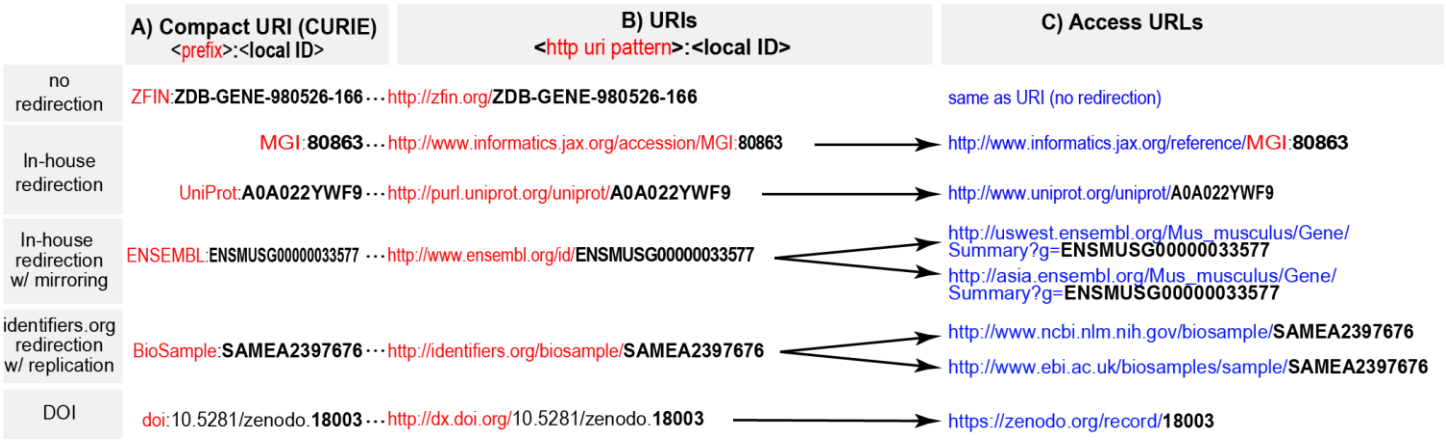

Fig. 2. Examples of provisioning resolvable URIs:
Compact URIs (CURIEs) (Panel A), URIs (Panel B) and Access URLs (Panel C) with no redirection (ZFIN), in house redirection (UniProt, and ENSEMBL), and 3rd party resolvers (using identifiers.org and DOI). In each case, the URI can be algorithmically derived from the CURIE because the **`Local ID`** portion itself is included (unmodified) within the URI. Access URL design patterns differ substantially by provider and may change over time. As long as `access URLs` are not used as the referenced ID, it does not matter whether they 1) contain a prefix and colon (MGI), or not (Ensembl), nor 2) whether they contain the entire `local ID` (Biosample) or not (DOI), nor 3) whether they include type (MGI) or not (ZFIN).

6. Implement a version-management policy

Changes in data resources impact how they can be referenced and used. Document your chosen version management practice: If you issue identifiers, document the change history for the resource (see also **Lesson 8**), or version the identifier itself, or do both.

Explicit identifier versioning is recommended if the prevailing use of an *unversioned* identifier results in "breaking changes" (e.g., a change in the hypothesized cause of a disease). However, if new information about the entity emerges slowly and the changes are "non-breaking", it is reasonable to instead maintain a machine-actionable change history wherein the changes are listed, and where they may also be categorized (eg. minor versus major changes). Versioning and change history work well together, especially when multiple types of changes overlap. Even where previous records are entirely removed, the URI should continue to resolve, but to a **"`tombstone`" page** (**Lesson 8**).

There are two approaches to versioning, record-level and release level; the latter is more common in the life sciences. If you version at the level of a database release, some reasonable number of prior versions of individual entities should still be resolvable via a versioned URI pattern eg. http://Jul2015.archive.ensembl.org/Mus_musculus/Gene/Summary?g=ENSMUSG00000033577, but preferably using an ISO date format or similarly deterministic convention.

If you version identifiers at the level of the individual record, you should version in the **`Local ID`** after the 'dot', as per UniProt in **Table 3**; this provides continuity in your site and also enables a single `prefix` to be used with any version: UniProt:P12345.3 ➐ http://www.uniprot.org/uniprot/P12345**.3.** If dot suffixing is not practicable, we strongly recommend providing a transparent mapping between identifiers as well as a mechanism for obtaining the latest version of the record. See Kratz et al. [24] for a more in-depth discussion of change management considerations.

Table 3. Recommendation for record-level versioning with URIs

Recommendation	**Example** (for clarity, **`Local ID`** only is shown)
Version information should follow after a dot	`P12345.3`
Base resource must resolve (302 redirect) to most recent version	`P12345`
Base resource should be deterministically convertible from version	`P12345.1` to `P12345`

Older versions must resolve	P12345.1
Illegal or invalid version must produce an informative HTTP error code (message such as 400 'Bad Request') and an HTML page explaining the error. For example identifiers.org has a page reporting "unknown collection" together with its 404 status.	P12345.302
Link from older version to current version must be provided	P12345.3
A list of all previous versions should be available	P12345 (see 'history' tab in user interface)
Two versions (or dates) should be comparable	http://www.uniprot.org/uniprot/P12345?version=*

7. Make URIs clear and findable

Make URIs obvious to users, especially where these differ from access URLs or application pages. For instance, at the record-level, advertise the "permanent link" together with a statement about persistence. E.g.

> *"The permanent link to this page, which will not change with the next release of Ensembl is: http://Jul2015.archive.ensembl.org/Mus_musculus/Gene/Summary?g=ENSMUSG00000033577;r=9:80165031-80311729;redirect=no We aim to maintain all archives for at least two years; some key releases may be maintained for longer"*

For archived records that are *out of date,* make this clear to the user and provide a link to the updated version (see http://www.uniprot.org/uniprot/P12345.1, for instance). Although it is good practice for each database website to include general citation guidance for users, it is increasingly important to provide a pre-populated citation *at the level of each record*. Outside of providers that issue DOIs, eagle-i[25] provides the best primary data source example of record-level citation instruction that we know of. Additional features that are useful in such widgets are that full references should be copy-pastable, integrated with reference managers, and pre-populated with the version information and access date.

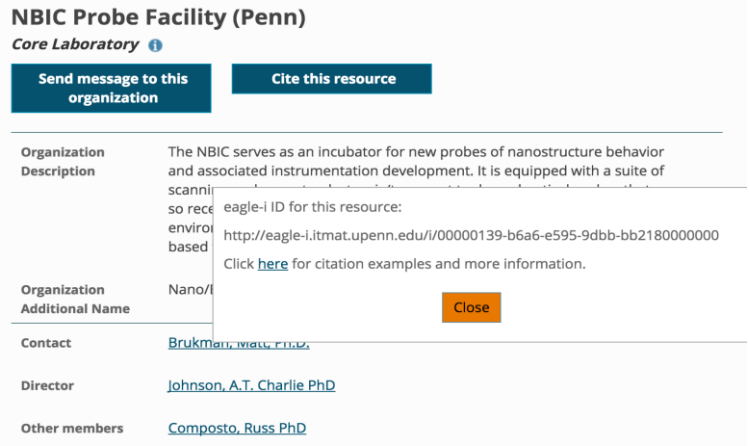

Figure 3. eagle-i record-level citation widget

8. Do not reassign or delete identifiers

Identifiers that have been exposed publicly, whether as HTTP URIs or via APIs may be deprecated but must never be deleted or reassigned to another record. If you issue identifiers, consider their full lifecycle: there is a fundamental difference between identifiers which point to experimental datasets (GenBank/ENA/DDBJ, PRIDE, etc.) and identifiers which point to a current understanding of a biological concept (Ensembl Gene, UniProt record, etc.). While experimental records are less likely to change, concept descriptions may evolve rapidly; even the nature and number of the relevant metadata fields change over time. Moreover, the very notion of identity is often strongly impacted by relationships (e.g., between concepts or processes).

Extensive changes cannot be captured with numerical suffixing alone. For instance, taxonomists may split or merge species, pathologists may split or merge diseases, or hypothesized entities may be proven not to exist

(e.g. vaccine-induced autism). Global initiatives (**Text S1**) are actively exploring identifier strategies for such use cases. In the meantime, consider **Table 4** recommendations.

Table 4. Recommendations for identifier lifecycle management

Recommended handling	Example
Obsoletion: If an entry has been removed or deprecated, the original identifier must still resolve to a `'tombstone page'`. Reasons for obsolescence should be indicated. If the obsoleted ID is replaced by another ID, the replacement must be present and also described as automatic or suggested, preferably using the ontology properties iao:`replaced_by` and `obo:consider`, respectively. The obsoleted ID must never be reassigned to another entity. A list of obsoleted IDs should be maintained.	Single obsoleted identifier: http://www.uniprot.org/uniprot/A0AV18 List of obsoleted identifiers: uniprot.org/help/deleted_accessions
Merging: When two or more identifiers are merged, a new recipient identifier should be designated as the primary (citable) one and should contain information about the legacy identifiers it encompasses. Any legacy identifiers should continue to resolve via redirection to the primary identifier.	UniProt entries Q57339 and O08022 have been merged into Q00626. Q57339 and O08022 are redirected to Q00626.
Splitting: If an identifier is split (demerged) into two or more new ones, new identifiers should be assigned to all the new entries. The legacy identifier must be marked as obsolete, but must also still resolve, providing a warning and pointers to the new ones as per above.	UniProt entry P29358 has been split into P68250 and P68251. P29358 displays a warning and links to the demerged entries: http://www.uniprot.org/uniprot/P29358

9. Document the identifiers you issue and use

The global-scale identification cycle is a shared responsibility and provider/consumer roles often overlap in the context of data integration. Whether you issue your own identifiers or just reference those of others, you should document your identifier policies. **Supplemental Table S5** provides a set of questions that data providers and re-distributors can use to develop such documentation. Documentation should be published alongside and/or included together in a dataset description, for instance, as outlined in the recommendations for Dataset Descriptions developed by the W3C Semantic Web in the Health Care and Life Sciences Interest Group [10]. For examples of such documentation see ChEMBL[26] and Monarch[27]; the format may vary.

10. Reference and display responsibly

The final lesson describes referencing recommendations for data redistributors: data aggregators, who collect information from different sources and re-display it; data publishers, who disseminate scientific knowledge through publications; and online reference material such as WikiData[28].

When external entities are referenced in narrative online text, they should be hyperlinked to their URIs or to pages/metadata containing their URIs. `Access URLs` are volatile (see **Lesson 4**) and must not be used for referencing or linking in any context intended to persist. Redistributors of data should monitor their references to other sources; any 'dead' links should be reported to the original data provider. If the original provider does not fix the broken link, your reference to it should be marked obsolete both visibly (for user interaction/interpretation), and within any accompanying metadata (for computational interaction/propagation). Differentiate identifiers linked internally within your application from identifiers linked outside your application; one way to do this is by using the linkout icon; consider opening all external links in a new browser window or tab in order to avoid confusion.

Broader issues associated with citation of data and software in the literature are outside of the scope of this paper, but **Text S1** lists relevant complementary efforts.

Conclusion

Better identifier design, provisioning, documentation, and referencing can address many of the identifier problems encountered in the life science data cycle. We recognize that improvements to the quality, diversity, and uptake of identifier tooling would lower barriers to adoption of these lessons. We will undertake to address these gaps in the relevant initiatives (**Text S1**). We also recognize the need for formal software-engineering specifications of identifier formats and/or alignment between existing specifications and hope that this paper can catalyze such efforts.

Supporting information

See attached

Acknowledgments

JA McMurry, T Burdett, N Juty, S Jupp, and C Morris were supported in part by the BioMedBridges project, which is funded by the European Union Seventh Framework Programme within Research Infrastructures of the FP7 Capacities Specific Programme, grant agreement number 284209. EMBL-EBI core funds supported H Parkinson, MJ Martin, J McEntyre, H Hermjakob, J Malone, M Courtot. ELIXIR core funds supported N Blomberg, R Jimenez. The European Commission provided additional support for Simon Jupp under grant number 601043 (“DIACHRON”) and for N Juty and H Hermjakob under grant number 312455 (“Infrastructure for Systems Biology - Europe (ISBE)”). The Drug Disease Model Resources grant number DDMoRe 115156 (“Innovative Medicines Initiative”) supported C. Laibe. Support was also received from the following BBSRC grants: BB/L005050/1 (“ELIXIR-UK, Manchester”) for SA Sansone, A Gonzalez-Beltran and C Goble; BB/M013189/1 (“DMMCore”) for C Goble, J Snoep, and N Stanford; BB/K019783/1 (“Continued development of ChEBI”) and BB/M006891/1 (“EMPATHY”) for N Swainston; BB/M017702/1 (“SYNBIOCHEM”) for N Swainson and D Fellows; BBS/E/B/000C0419 (“A systems approach to understanding lipid, Ca2+ and MAPK signalling networks”) for N Le Novère; BB/L005069/1 (“ELIXIR-UK, Oxford”) for SA Sansone, A Gonzalez-Beltran and P Rocca-Serra. NIH support was provided from the following grants: U41HG007822 (“UniProt”) for MJ Martin; U24AI117966-01 (“bioCADDIE”) for SA Sansone, A Gonzalez-Beltran and P Rocca-Serra; U54AI117925 (“CEDAR”) for M Dumontier, SA Sansone, A Gonzalez-Beltran and P Rocca-Serra; R24OD011883 (“Monarch Initiative”) for CJ Mungall, MA Haendel, JA McMurry and NL Washington; NHGRI P41HG002273-09 (“Gene Ontology Consortium”) for CJ Mungall. Additional support for CJ Mungall and NL Washington was received from the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under [Contract No. DE-AC02-05CH11231].

The authors wish to thank Mary Todd Bergman, Ewan Birney, Fiona Cunningham, Richard Cyganiak, Adam Faulconbridge, Andrew M Jenkinson,Sirarat Sarntivijai, Stephanie Suhr, Eleanor Williams, and Tim Clark for their valuable feedback and suggestions. We also wish to thank the BioMedBridges Scientific Advisory Board for the suggestion to address this important issue and the reviewers for their constructive comments.

References

1. Van de Sompel H, Sanderson R, Shankar H, Klein M. Persistent Identifiers for Scholarly Assets and the Web: The Need for an Unambiguous Mapping. International Journal of Digital Curation 2014; 9: 331-342. doi: 10.2218/ijdc.v9i1.320

2. Guralnick RP, Cellinese N, Deck J, Pyle RL, Kunze J, Penev L, Walls R, Hagedorn G, Agosti D, Wieczorek J, Catapano T, Page EDM. Community Next Steps for Making Globally Unique Identifiers Work for Biocollections Data. ZooKeys 2015; 494: 133-154. doi:10.3897/zookeys.494.9352

3. Data Citation Synthesis Group. Joint Declaration of Data Citation Principles. Martone M. (ed.) San Diego CA: FORCE11. 2013. Available from: https://www.force11.org/datacitation

4. Altman M, Crosas M. The Evolution of Data Citation: From Principles to Implementation. IASSIST Quarterly 37. 2013. Available from: http://www.iassistdata.org/downloads/iqvol371_4_altman.pdf

5. Bandrowski A, Brush M, Grethe JS, Haendel MA, Kennedy DN, Hill S, et al. The Resource Identification Initiative: A cultural shift in publishing [version 2; referees: 2 approved]. F1000Research 2015, 4:134 doi: 10.12688/f1000research.6555.2

6. JDDCP (Joint Declaration of Data Citation Principles). The FAIR data Guiding Principles. 2014. Available from: https://www.force11.org/group/fairgroup/fairprinciples.

7. Starr J, Castro E, Crosas M, Dumontier M, Downs RR, Duerr R, et al. Achieving human and machine accessibility of cited data in scholarly publications. *PeerJ Computer Science* 2015; 1:e1. doi:10.7717/peerj-cs.1

8. Bradner S. Key words for use in RFCs to Indicate Requirement Levels. 1997. Available from: http://www.ietf.org/rfc/rfc2119.txt, doi:10.17487/RFC2119

9. European Bioinformatics Institute. Protein Identifiers Cross-Reference REST documentation. 2015. Available from: http://www.ebi.ac.uk/Tools/picr/RESTDocumentation.do

10. Gray AJG, Baran J, Marshall MS, Dumontier M. (eds). Dataset Descriptions: HCLS Community Profile. W3C Interest Group Note 14 May 2015. Available from: https://www.w3.org/TR/2015/NOTE-hcls-dataset-20150514/#s6_3.

11. Mietchen D, McEntyre J, Beck J, et al. Force11 Data Citation Implementation Group Adapting JATS to support data citation. In: Journal Article Tag Suite Conference (JATS-Con) Proceedings. 2015. Available from: http://www.ncbi.nlm.nih.gov/books/NBK280240/?report=classic

12. Van de Sompel, H., Nelson, M., and R. Sanderson, "HTTP Framework for Time-Based Access to Resource States -- Memento", RFC 7089, 2013, DOI:10.17487/RFC7089, http://www.rfc-editor.org/info/rfc7089.

13. Rafael C Jimenez, Alejandra Gonzalez-Beltran, Martin Cook, Niall Beard, Pistoia Alliance, Roberto Preste, (2016) Bioschemas: promoting consistent adoption of schema.org markup within the life sciences. Available from: http://bioschemas.org.

14. Guralnick RP, Cellinese N, Deck J, Pyle RL, Kunze J, Penev L, Walls R, Hagedorn G, Agosti D, Wieczorek J, Catapano T, Page EDM. Community Next Steps for Making Globally Unique Identifiers Work for Biocollections Data. ZooKeys. 2015; 494: 133-154. doi: 10.3897/zookeys.494.9352

15. Berners-Lee T. Cool URIs don't change. 1998. Available from: http://www.w3.org/Provider/Style/URI.html.

16. Klein M, Van de Sompel H, Sanderson R, Shankar H, Balakireva L, Zhou K, et al. Scholarly context not found: one in five articles suffers from reference rot. PLoS One. 2014; 9: e115253. doi: 10.1371/journal.pone.0115253.

17. Sporny M. The Case for Curies. 2011. Available from: http://manu.sporny.org/2011/case-for-curies/.

18. Birbeck M, McCarron S (eds). CURIE Syntax 1.0. W3C Working Group Note. 16 December 2010. Available from: http://www.w3.org/TR/2010/NOTE-curie-20101216.

19. Mungall C, McMurry JA. 2015. PrefixCommons BioContext: JSON-LD Contexts for Bioinformatics Data. Available from: https://github.com/prefixcommons/biocontext.

20. Berners Lee, T. The Need for a Universal Syntax. 1993. Available from: http://www.w3.org/Addressing/URL/uri-spec.html

21. Zeeberg BR, Riss J, Kane DW, Bussey KJ, Uchio E, Linehan WM, et al. Mistaken identifiers: gene name errors can be introduced inadvertently when using Excel in bioinformatics. BMC Bioinformatics. 2004; 23: 80. doi:10.1186/1471-2105-5-80

22. Mark Ziemann, Yotam Eren and Assam El-Osta. Gene name errors are widespread in the scientific literature. Genome Biology. 2016; 17:177. DOI: 10.1186/s13059-016-1044-7

23. Kunze J, Rodgers R. ARK Specification. 2016. Available from: http://www.cdlib.org/services/uc3/arkspec.pdf, https://wiki.ucop.edu/display/Curation/ARK

24. Kratz J, Strasser C. Data publication consensus and controversies [v3; ref status: indexed, http://f1000r.es/4ja] F1000Research. 2014; 3:94 doi: 10.12688/f1000research.3979.3

25. Vasilevsky N, Johnson T, Corday K, Torniai C, Brush M, Segerdell E, Wilson M, Shaffer C, Robinson D, Haendel M. Research resources: curating the new eagle-i discovery system. Database (Oxford). 2012; bar067. doi: 10.1093/database/bar067

26. Complete Example of a Dataset Description. 2015. http://www.w3.org/TR/hcls-dataset/#appendix_1. Available from: http://www.w3.org/TR/hcls-dataset/#appendix_1

27. McMurry J, Washington N, Shefcheck K, Conlin T. 2015. DIPPER: The Monarch Data Ingest Pipeline Identifier Documentation. Available from: https://github.com/monarch-initiative/dipper/blob/master/README.md#identifiers

28. Wikimedia Foundation. WikiData: a free, collaborative, multilingual, secondary database, collecting structured data to provide support for Wikipedia, Wikimedia Commons, the other Wikimedia projects, and well beyond that. 2015. Available from: https://www.wikidata.org/