Database 1

Biological databases and ontology

Cheong Xin Chan (CX)

c.chan1@uq.edu.au

Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics 1 – Introduction

Outline

- **Biological data and data types**
- **Biological databases**
  - Main functions and applications in bioinformatics
  - Characteristics and types of biological databases
  - Issues and challenges
- **Organisation of biological data and Ontology**
  - Definition and rationales
  - Basic terminology and types of ontologies
  - Linking data in Semantic Web
- **Biological ontologies**
  - Development and resources of biological ontologies
  - Gene Ontology and other examples
  - Issues and challenges

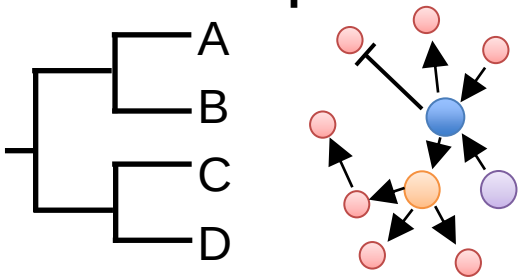
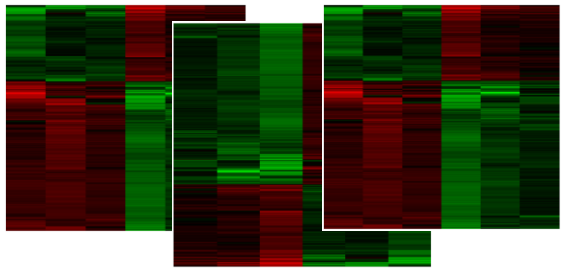
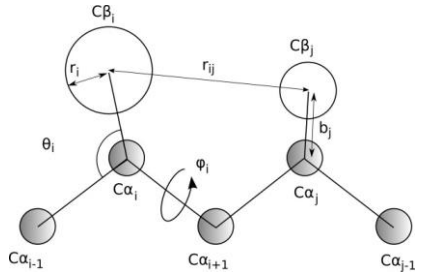
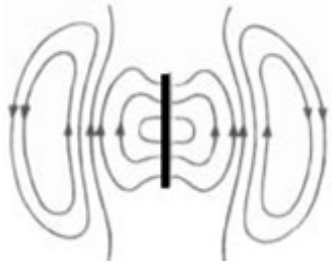
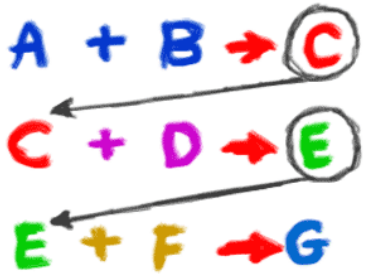
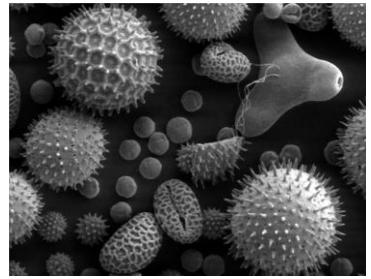
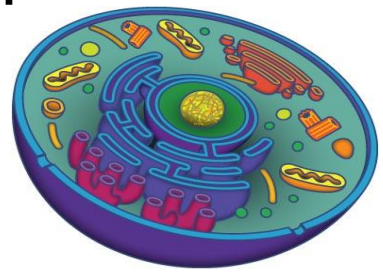
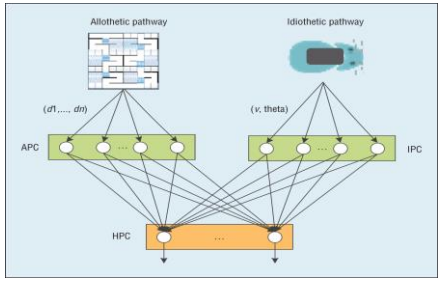
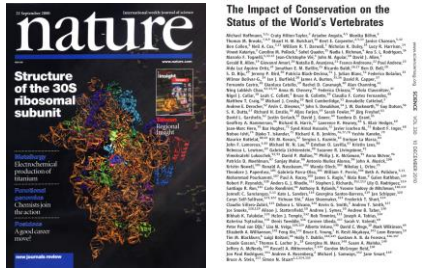
Biological data

include:

(a) information or measurements generated from biological resources and/or experiments (i.e. the **data**), and

(b) information that describes and/or related to characteristics/features of a biological entity/an experiment (i.e. the **metadata**, or the *data* about the data)

Biological data types

Sequences CGACTACGATCAGCTA CGACGACTACGATCAG CATGCATCACAGCTAG CATCGACTAGCATCGA TCACGACTAGCGCATG	**Graphs** 	**High-dimensional data** 	**Geometric information** 
Scalar & vector fields 	**Patterns** 	**Constraints** 	**Images** 
Spatial information 	**Models** 	**Prose/Literature** 	**Declarative knowledge (hypotheses, evidence)**

Data on a biological entity can be associated with one or more of these types: e.g., a protein might have associated with it two-dimensional images, three-dimensional structures, one-dimensional sequences, annotations of these data structures, etc.

Biological data in the digital world

- Biological data are **heterogeneous**
- Ideally these data should be **shareable** and **interoperable** among diverse laboratories and computer systems
- Most data are now in digital, machine-readable forms
- Common digital formats of biological data (non-exhaustive): en.wikipedia.org/wiki/Biological_data

Example 1: sequences

Sequence data

Multiple sequence alignment

```
>sequence1
MDSKGSSQKGSRLLLLLVVSNLLLCQGVVSTPVCPNGPGNCQVSLRDLFDRAVMVSHYIHDLSS
EMFNEFDKRYAQGKGFITMALNSCHTSSLPTPEDKEQAQQTHHEVLMSLILGLLRSWNDPLYHL
VTEVRGMKGAPDAILSRAIEIEEENKRLLEGMEMIFGQVIPGAKETEPYPVWSGLPSLQTKDED
ARYSAFYNLLHCLRRDSSKIDTYLKLLNCRIIYNNNC*
>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken
ADQLTEEQIAEFKEAFSLFDKDGDGTITTKELGTVMRSLGQNPTEAELQDMINEVDADGNGTID
FPEFLTMMARKMKDTDSEEEIREAFRVFDKDGNGYISAAELRHVMTNLGEKLTDEEVDEMIREA
DIDGDGQVNYEEFVQMMTAK*
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPY
IGTNLVEWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIP
FHPYYTIKDFLGLLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRS
VPNKLGGVLALFLSIVILGLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPY
TIIGQMASILYFSIILAFLPIAGXIENY
```

FASTA format

FASTA format

```
>Seq5541
LNPSAPTVVVLGWLGASIKHLAKY
>Seq7032
QKPKRPTAVLLGWFAAKHKNLSKY
>Seq7152
MRPARGFVVVLGWFGAQDKHLKKY
>Seq2877
LVPWAPTAVLLGWVGCQMRYLRKY
>Seq0056
----RPLVLTLGWLGANERHLGKY
>Seq0781
GFLLNPLVIVMGWHGCKPRYLSKY
>Seq2239
QDPASVIVVLLGWYACHPKVLAKY
>Seq5612
KFPKVPIVMLLGWAGCQDRYLMKY
>Seq4904
VFSEEPVVILLGWAGSRDKHLAKY
>Seq5924
EIPDLPLVILLGWGGCSDKNLAKY
>Seq7619
EIPDQPVVILLGWGGCRDKNLAKY
```

PHYLIP format

```
 11 24
Seq5541 LNPSAPTVVVLGWLGASIKHLAKY
Seq7032 QKPKRPTAVLLGWFAAKHKNLSKY
Seq7152 MRPARGFVVVLGWFGAQDKHLKKY
Seq2877 LVPWAPTAVLLGWVGCQMRYLRKY
Seq0056 ----RPLVLTLGWLGANERHLGKY
Seq0781 GFLLNPLVIVMGWHGCKPRYLSKY
Seq2239 QDPASVIVVLLGWYACHPKVLAKY
Seq5612 KFPKVPIVMLLGWAGCQDRYLMKY
Seq4904 VFSEEPVVILLGWAGSRDKHLAKY
Seq5924 EIPDLPLVILLGWGGCSDKNLAKY
Seq7619 EIPDQPVVILLGWGGCRDKNLAKY
```

NEXUS format

```
#NEXUS
[MySeqData.phy -- data title]

[Name: Seq5541          Len:    24  Check: 3190F9FF]
[Name: Seq7032          Len:    24  Check: 5463206F]
[Name: Seq7152          Len:    24  Check: 1CC258CA]
[Name: Seq2877          Len:    24  Check: D21B9C87]
[Name: Seq0056          Len:    24  Check: CF841852]
[Name: Seq0781          Len:    24  Check:  A625362]
[Name: Seq2239          Len:    24  Check: 9F481732]
[Name: Seq5612          Len:    24  Check: AC34C3CF]
[Name: Seq4904          Len:    24  Check: 817ABE64]
[Name: Seq5924          Len:    24  Check: C2CBAE73]
[Name: Seq7619          Len:    24  Check: C2CBAE73]


begin data;
 dimensions ntax=11 nchar=24;
 format datatype=protein interleave missing=-;
  matrix
   Seq5541  LNPSAPTVVVLGWLGASIKH LAKY
   Seq7032  QKPKRPTAVLLGWFAAKHKN LSKY
   Seq7152  MRPARGFVVVLGWFGAQDKH LKKY
   Seq2877  LVPWAPTAVLLGWVGCQMRY LRKY
   Seq0056  ----RPLVLTLGWLGANERH LGKY
   Seq0781  GFLLNPLVIVMGWHGCKPRY LSKY
   Seq2239  QDPASVIVVLLGWYACHPKV LAKY
   Seq5612  KFPKVPIVMLLGWAGCQDRY LMKY
   Seq4904  VFSEEPVVILLGWAGSRDKH LAKY
   Seq5924  EIPDLPLVILLGWGGCSDKN LAKY
   Seq7619  EIPDQPVVILLGWGGCRDKN LAKY

;
  end;
```

Example 2: sequence records

```
LOCUS       HG941718             5109767 bp    DNA     circular BCT 03-APR-2015
DEFINITION  Escherichia coli ST131 strain EC958 chromosome, complete genome.
ACCESSION   HG941718
VERSION     HG941718.1  GI:641682562
DBLINK      BioProject: PRJEA61443
            BioSample: SAMEA2272019
KEYWORDS    complete genome.
SOURCE      Escherichia coli O25b:H4-ST131
  ORGANISM  Escherichia coli O25b:H4-ST131
            Bacteria; Proteobacteria; Gammaproteobacteria; Enterobacteriales;
            Enterobacteriaceae; Escherichia.
REFERENCE   1
  AUTHORS   Forde,B.M., Ben Zakour,N.L., Stanton-Cook,M., Phan,M.D.,
            Totsika,M., Peters,K.M., Chan,K.G., Schembri,M.A., Upton,M. and
            Beatson,S.A.
  TITLE     The complete genome sequence of Escherichia coli EC958: a high
            quality reference sequence for the globally disseminated multidrug
            resistant E. coli O25b:H4-ST131 clone
  JOURNAL   PLoS ONE 9 (8), E104400 (2014)
  PUBMED    25126841
  REMARK    Publication Status: Online-Only
REFERENCE   2  (bases 1 to 5109767)
  AUTHORS   Beatson,S.
  TITLE     Direct Submission
  JOURNAL   Submitted (05-OCT-2011) The University of Queensland, Centre for
            Infectious Disease Research, St. Lucia, Brisbane, QLD 4072,
            AUSTRALIA
FEATURES             Location/Qualifiers
     source          1..5109767
                     /organism="Escherichia coli O25b:H4-ST131"
                     /mol_type="genomic DNA"
                     /strain="EC958"
                     /serotype="O25b:H4"
                     /db_xref="taxon:941322"
...
```

GenBank format (NCBI USA)

```
ID   HG941718; SV 1; circular; genomic DNA; STD; PRO; 5109767 BP.
AC   HG941718;
PR   Project:PRJEA61443;
DT   11-MAR-2014 (Rel. 120, Created)
DT   03-APR-2015 (Rel. 124, Last updated, Version 6)
DE   Escherichia coli ST131 strain EC958 chromosome, complete genome
KW   complete genome.
OS   Escherichia coli ST131
OC   Bacteria; Proteobacteria; Gammaproteobacteria; Enterobacteriales;
OC   Enterobacteriaceae; Escherichia.
RN   [1]
RP   1-5109767
RA   Beatson S.;
RT   ;
RL   Submitted (05-OCT-2011) to the INSDC.
RL   The University of Queensland, Centre for Infectious Disease Research, St.
RL   Lucia, Brisbane, QLD 4072, AUSTRALIA.
RN   [2]
RX   DOI; 10.1371/journal.pone.0104400.
RX   PUBMED; 25126841.
RA   Forde B.M., Ben Zakour N.L., Stanton-Cook M., Phan M.D., Totsika M.,
RA   Peters K.M., Chan K.G., Schembri M.A., Upton M., Beatson S.A.;
RT   "The complete genome sequence of Escherichia coli EC958: a high quality
RT   reference sequence for the globally disseminated multidrug resistant E.
RT   coli O25b:H4-ST131 clone";
RL   PLoS One 9(8):e104400-e104400(2014).
...
FH   Key             Location/Qualifiers
FH
FT   source          1..5109767
FT                   /organism="Escherichia coli ST131"
FT                   /strain="EC958"
FT                   /serotype="O25b:H4"
FT                   /mol_type="genomic DNA"
FT                   /db_xref="taxon:1359206"
...
```

EMBL format (EBI Europe)

- **Standardised definitions** within a format – these can be format-specific
- **Cross-references** allow for integration of information from different databases

Example 3: protein structures

```
HEADER    TRANSFERASE                             29-JUL-07   2Z6C
TITLE     CRYSTAL STRUCTURE OF LOV1 DOMAIN OF PHOTOTROPIN1 FROM
TITLE    2 ARABIDOPSIS THALIANA
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: PHOTOTROPIN-1;
COMPND   3 CHAIN: A, B;
COMPND   4 FRAGMENT: UNP RESIDUES 180-308, LOV1 DOMAIN;
COMPND   5 SYNONYM: NON-PHOTOTROPIC HYPOCOTYL PROTEIN 1, ROOT
COMPND   6 PHOTOTROPISM PROTEIN 1;
COMPND   7 EC: 2.7.11.1;
COMPND   8 ENGINEERED: YES
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: ARABIDOPSIS THALIANA;
SOURCE   3 ORGANISM_COMMON: MOUSE-EAR CRESS;
SOURCE   4 ORGANISM_TAXID: 3702;
SOURCE   5 GENE: PHOT1, JK224, NPH1, RPT1;
SOURCE   6 EXPRESSION_SYSTEM: ESCHERICHIA COLI;
SOURCE   7 EXPRESSION_SYSTEM_TAXID: 562
...
ATOM      1  N   VAL A 184      63.131  56.497  -7.951  1.00 66.06           N
ATOM      2  CA  VAL A 184      63.000  57.402  -6.758  1.00 66.09           C
ATOM      3  C   VAL A 184      61.746  58.281  -6.840  1.00 65.59           C
ATOM      4  O   VAL A 184      60.942  58.285  -5.910  1.00 65.83           O
ATOM      5  CB  VAL A 184      64.308  58.238  -6.490  1.00 66.38           C
ATOM      6  CG1 VAL A 184      64.019  59.583  -5.767  1.00 67.05           C
ATOM      7  CG2 VAL A 184      65.335  57.397  -5.699  1.00 67.19           C
ATOM      8  N   SER A 185      61.574  59.014  -7.941  1.00 64.93           N
ATOM      9  CA  SER A 185      60.374  59.839  -8.108  1.00 64.24           C
ATOM     10  C   SER A 185      59.090  59.005  -8.233  1.00 63.86           C
ATOM     11  O   SER A 185      58.967  58.134  -9.107  1.00 63.28           O
ATOM     12  CB  SER A 185      60.513  60.821  -9.276  1.00 64.53           C
...
```

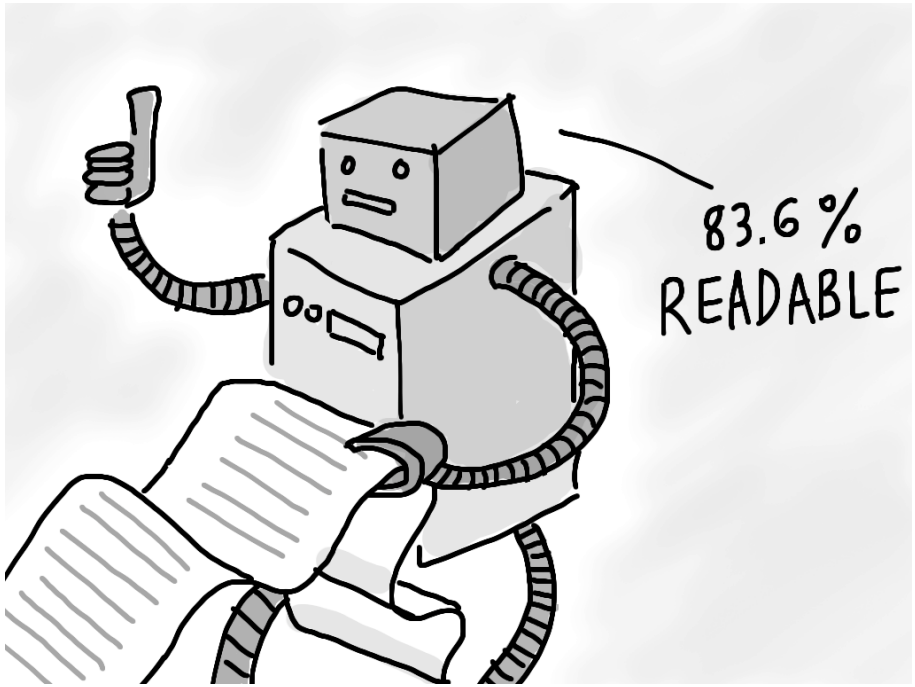
PDB format – the standard representation for macromolecular structure data

Biological databases

Two main functions:

Emilio Quintana (https://www.flickr.com/photos/eq/)

Make biological data available to scientists



UXMastery.com

Make biological data available in computer-readable form

Applications of databases in bioinformatics (and modern molecular biosciences)

Identification of biological entities

- genes, proteins, metabolites, reactions etc.
- homologs, orthologs, gene/protein families

Inference of function

- sequence/structural similarity, prediction or association

Hypothesis generation

- Source data for experiments/to train and test tools, models and methods

Characteristics of biological databases

Type of data	Data entry & quality control	Primary or derived data
• Nucleotide • Protein • Macromolecular structure • Gene expression • Metabolic pathways • Protein-protein interactions • …	• Scientists (teams) deposit data directly • Appointed curators add and update data • Are erroneous data removed or marked? • Type and degree of error checking • Consistency, redundancy, conflicts, updates	• Primary databases: experimental results directly into database • Secondary databases: results of analysis of primary databases • Aggregate of many databases • Links to other data items • Combination of data • Consolidation of data
Technical design	**Maintainer status**	**Availability**
• Flat-files • Relational database (MySQL) • Object-oriented database (e.g. XML, JSON formats, PostgreSQL)	• Large, public institution (e.g. EMBL, NCBI) • Quasi-academic institute (e.g. Swiss Institute of Bioinformatics, J. Craig Venter Institute) • Academic group or scientist • Commercial company	• Publicly available, no restrictions • Available, but with copyright • Accessible, but not downloadable • Academic, but not freely available • Proprietary, commercial; possibly free for academics

One or more of these factors affect which database resources does one use, and how.

Types of biological databases:

Sequence databases

What can you get?

- Sequences, from whole genomes to protein isoforms
- Biomolecular and chemical structures
- Functional annotation
- Analysis tools

What can you do with them?

- Homology inference
- Phylogenetic analysis
- Sequence-based feature prediction, e.g. conserved patterns (motifs)
- Functional analysis

International Nucleotide Sequence Database Collaboration (INSDC)

- GenBank www.ncbi.nlm.nih.gov
- EMBL-EBI www.ebi.ac.uk
- DNA DataBank of Japan www.ddbj.nig.ac.jp

Mandatory sequence submission to public repositories (making data freely available) prior to scientific publications

Growth of sequence databases

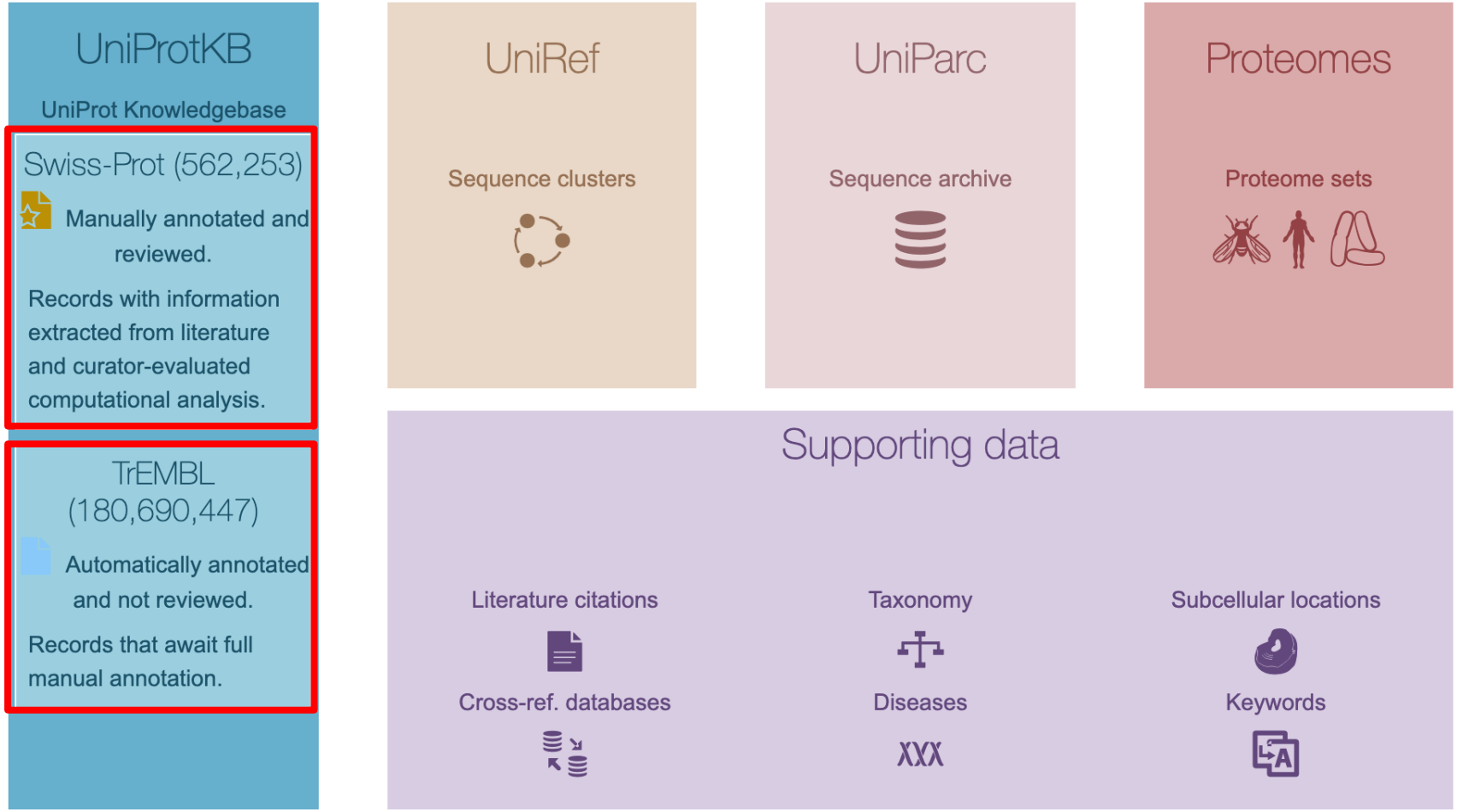
1960s	• First Atlas of Protein Sequence and Structure; Vol. 1 (1965) • Vol. 4 (1969): > 300 protein sequences; 16 DNA sequences
1970s	• Recognition of the role of databases in collecting & managing sequence data • Efforts to make sequence data available to biologists
1980s	• Sequence databases established by NIH and EMBL • DDBJ/EMBL/GenBank (1982); **606** nucleotide seqs (Dec 1982)
1990s	• Internet • Increase in sequence submission to public repositories
2000s	• Continued growth of sequence data including whole genomes
2010s+	• Challenges posed by high-throughput sequencing • **1.79** billion nucleotide sequences in GenBank including WGS (Feb 2021)

Types of biological databases:

Annotation databases

UniProt Knowledgebase (UniProtKB):

www.uniprot.org



Cross-referencing to other databases

- Identifiers, alternative names, accession numbers

Annotation

- Description of function
- Isoforms and sequences
- Localisation
- Post translational modifications
- Domains, motifs, signal sequences
- Interactions

Useful for

- Collecting sets of functionally related proteins (e.g. families)
- Identifying protein features

Annotation databases

Gene Ontology database (geneontology.org)

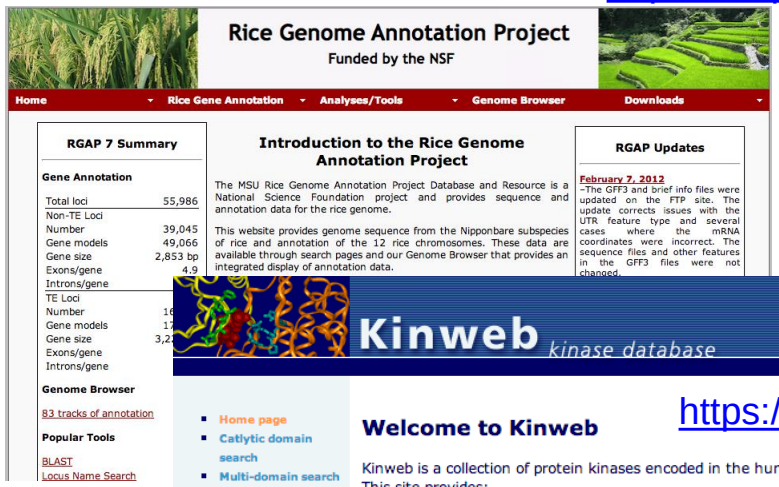
- Unified attributes of gene and gene product across all species
- Maintained by the GO Consortium

Types of biological databases:

Context-specific databases

- Established and maintained by the research communities
- Species-specific
- Disease-specific
- System-specific
  e.g. kinases (Kinweb), nuclear receptors (NURSA), allosteric molecules (ASD), membrane transporters (transportDB)

http://rice.plantbiology.msu.edu/



https://www.itb.cnr.it/kinweb/

http://mdl.shsmu.edu.cn/ASD/

Advantages

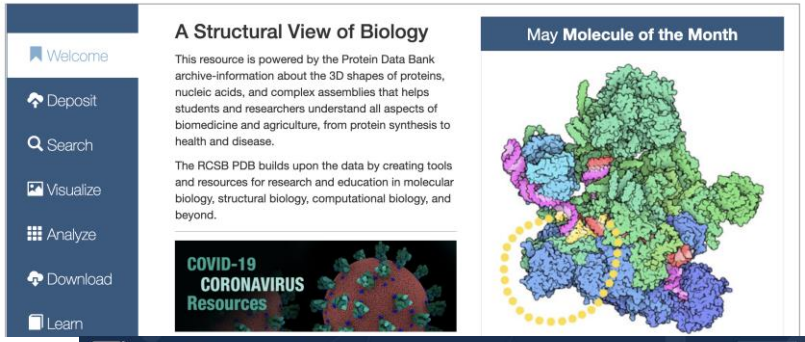
- Maintained by the research community
- Data curation by expert community
- Detailed, high-quality annotation

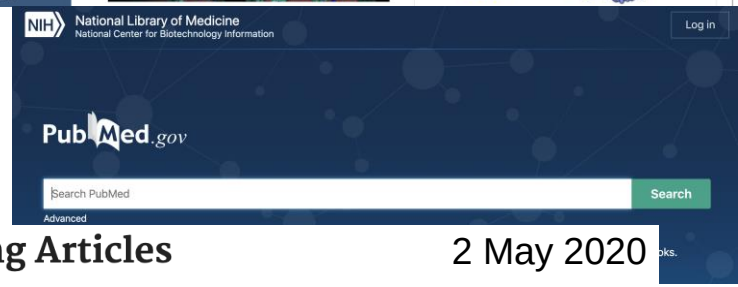
Disadvantages

- Maintained by the research community
- Idiosyncratic naming conventions
- Poor mapping to external DBs
- Sporadic updates

Other context-specific databases

- **InterPro:** www.ebi.ac.uk/interpro
  - domain models of proteins from ~20 primary resources
- **Protein Data Bank (PDB):** www.rcsb.org/pdb/
  - high-resolution 3D structural data of proteins
- **Gene Expression Omnibus (GEO):** www.ncbi.nlm.nih.gov/geo/
  ArrayExpress: www.ebi.ac.uk/arrayexpress/
  - high-throughput functional genomics (gene expression) data
- **Oncomine:** www.oncomine.org
  - cancer microarray data & web-based data-mining platform
- **SymbioGBR:** http://www.symbiogbr.org/
  - database of coral symbionts from the Great Barrier Reef
- **KEGG Pathway Database:** www.genome.jp/kegg/pathway.html
  - pathways of molecular interaction and reaction networks for metabolism, information processes, cellular processes, organismal systems & human diseases
- **PubMed (NCBI):** www.ncbi.nlm.nih.gov/pubmed/
  - scientific literature relevant to biomedical fields
- and many more …





Trending Articles 2 May 2020

PubMed records with recent increases in activity

COVID-19 diagnosis and management: a comprehensive review.
Pascarella G, et al. J Intern Med. 2020. PMID: 32348588

Hypercoagulation and Antithrombotic Treatment in Coronavirus 2019: A New Challenge.
Violi F, et al. Thromb Haemost. 2020. PMID: 32349133

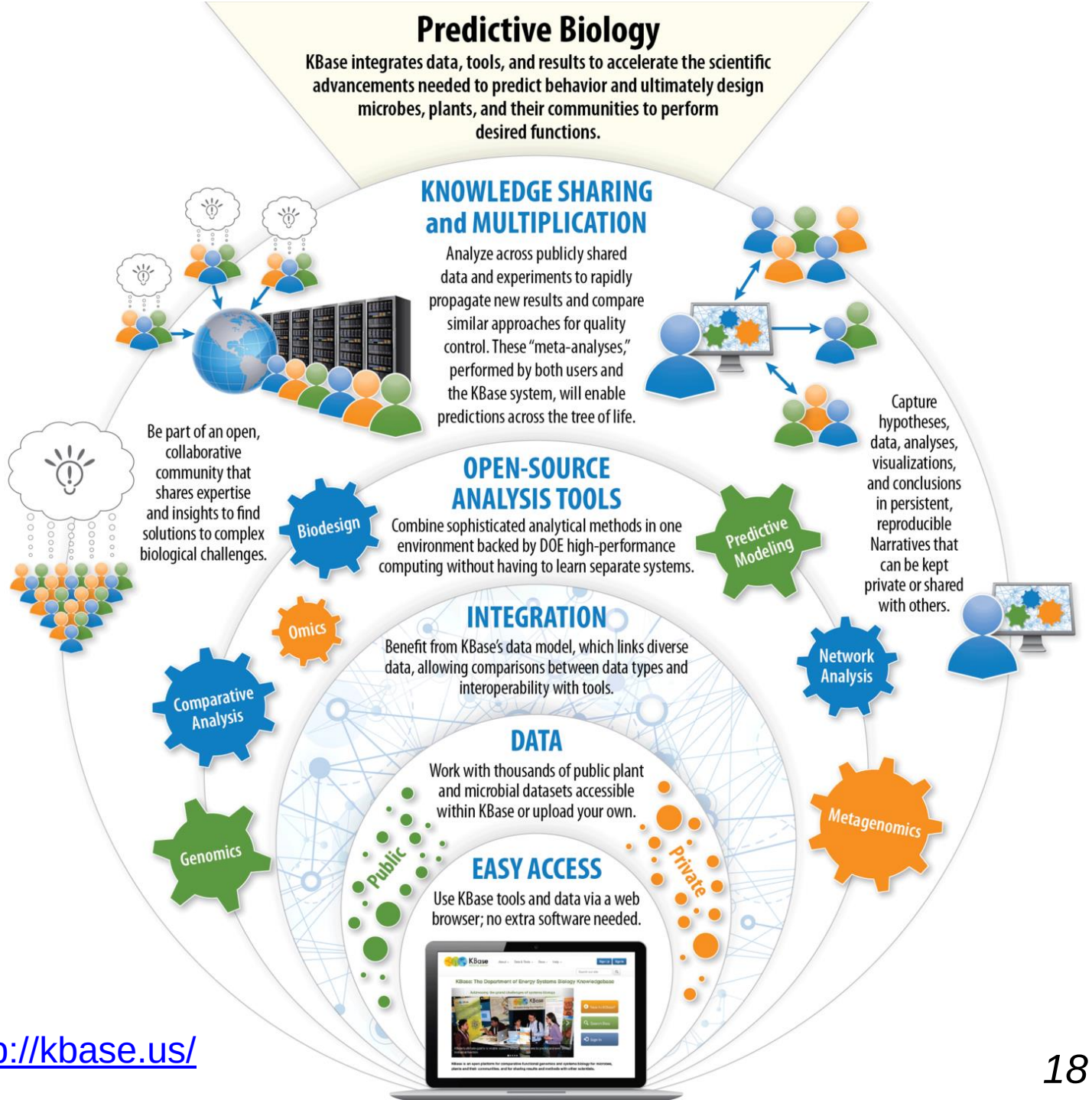
A SARS-CoV-2 protein interaction map reveals targets for drug repurposing.
Gordon DE, et al. Nature. 2020. PMID: 32353859

DIC in COVID-19: Implications for Prognosis and Treatment?
Seitz R, et al. J Thromb Haemost. 2020. PMID: 32344469

Finding the dose for hydroxychloroquine prophylaxis for COVID-19; the desperate search for effectiveness.
Al-Kofahi M, et al. Clin Pharmacol Ther. 2020. PMID: 32344449

Databases as a knowledgebase

- integrates and processes data (usually from multiple databases) and uses expert knowledge to give answers, recommendations and expert advice
- one-stop platform allows for multi-scale modelling, data integration, collaboration and sharing



A workflow example:
U.S. Department of Energy's Systems Biology Knowledgebase (KBase) http://kbase.us/

Biological databases: Issues and challenges

Data heterogeneity

- various data types & data quality impact the ease of curation, automation and integration

Proprietary issues

- private databases are not readily accessible to the research community

Disparate terminology

- standardisation of terms or formats through time

Shareability & interoperability

- interfaces for data exchange & data-format description, interfaces to recognise data-model intersections, to exchange metadata and to parse queries

Organisation of biological data & ontology

- Data are useless if they are inaccessible or incomprehensible to others
- Data organisation is central to extracting useful information from the data
- Aim: one large, distributed information resource with **common controlled vocabularies**, related **user interfaces**, and **practices**
- Emphasis: **shareability** and **interoperability** of large-scale **heterogeneous** data

Ontology

Greek origin: *ontos* – being or the nature of things; *logia* – science, study, theory

- defines (specifies) the **concepts**, **relationships**, and other distinctions that are relevant for modeling a domain
- takes the form of the definitions of **representational vocabulary** (classes, relations etc.), which provide meanings for the vocabulary and **formal constraints** on its coherent use
- commonly based on agreed-upon understanding of a domain, i.e. a **joint terminology** between members of a community of interest
- a vocabulary of concepts and relations rich enough to enable us to express knowledge and intention **without semantic ambiguity**

Why do we need ontology?

- to **share common understanding** of the structure of information among people or software agents
- to **enable reuse** of domain knowledge
- to make domain **assumptions explicit**

Natalya Noy & Deborah McGuinness, Stanford University:
http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

Basic terminology

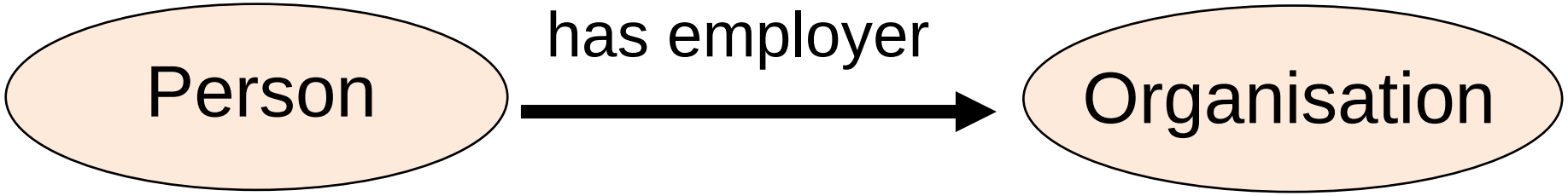
Class

Person

Organisation

Relations

has employer,
is employee of
….



Instance (Individual)

Triple

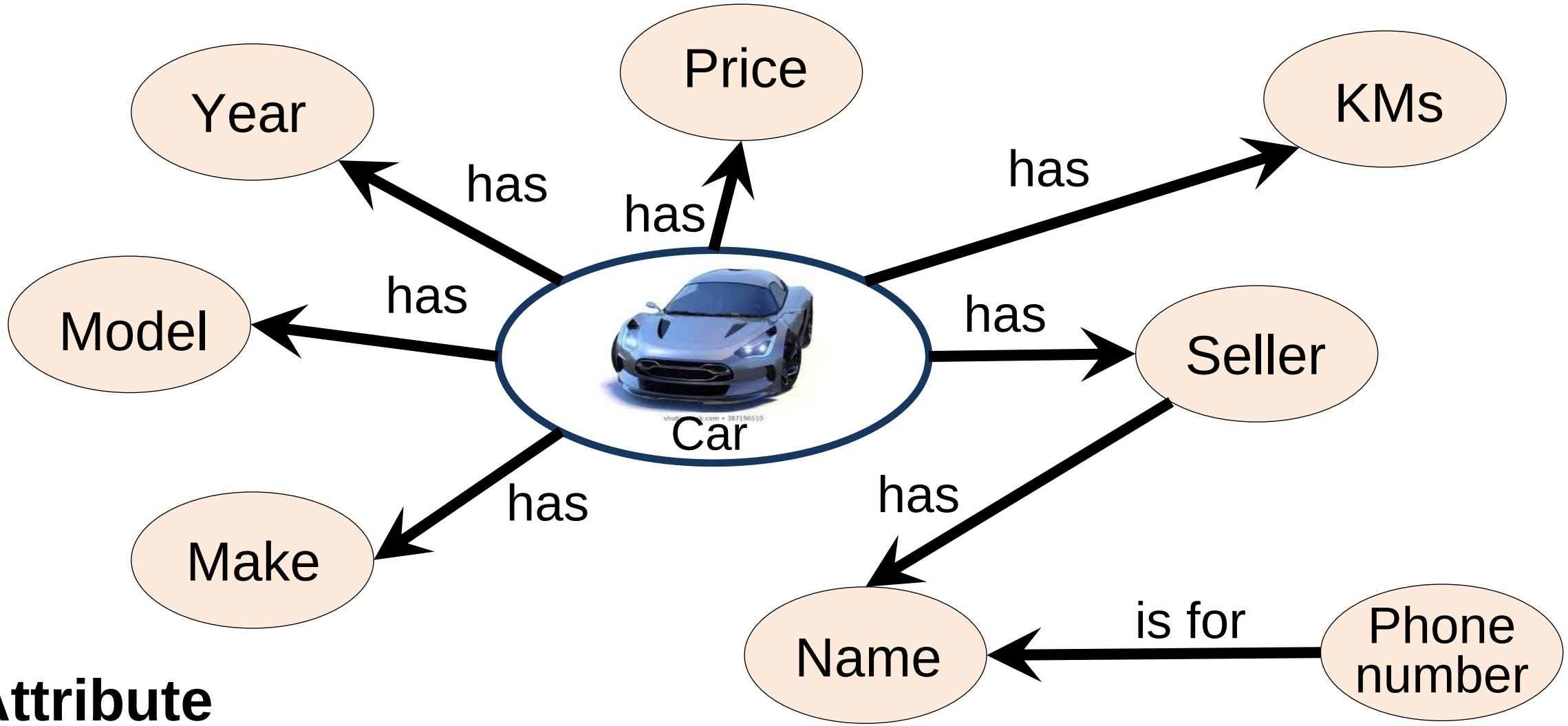
Subject
Andrew

Predicate
has employer

Object
University of Queensland

https://www.youtube.com/watch?v=jfUPLuPL3Ho

Example: a car advertisement

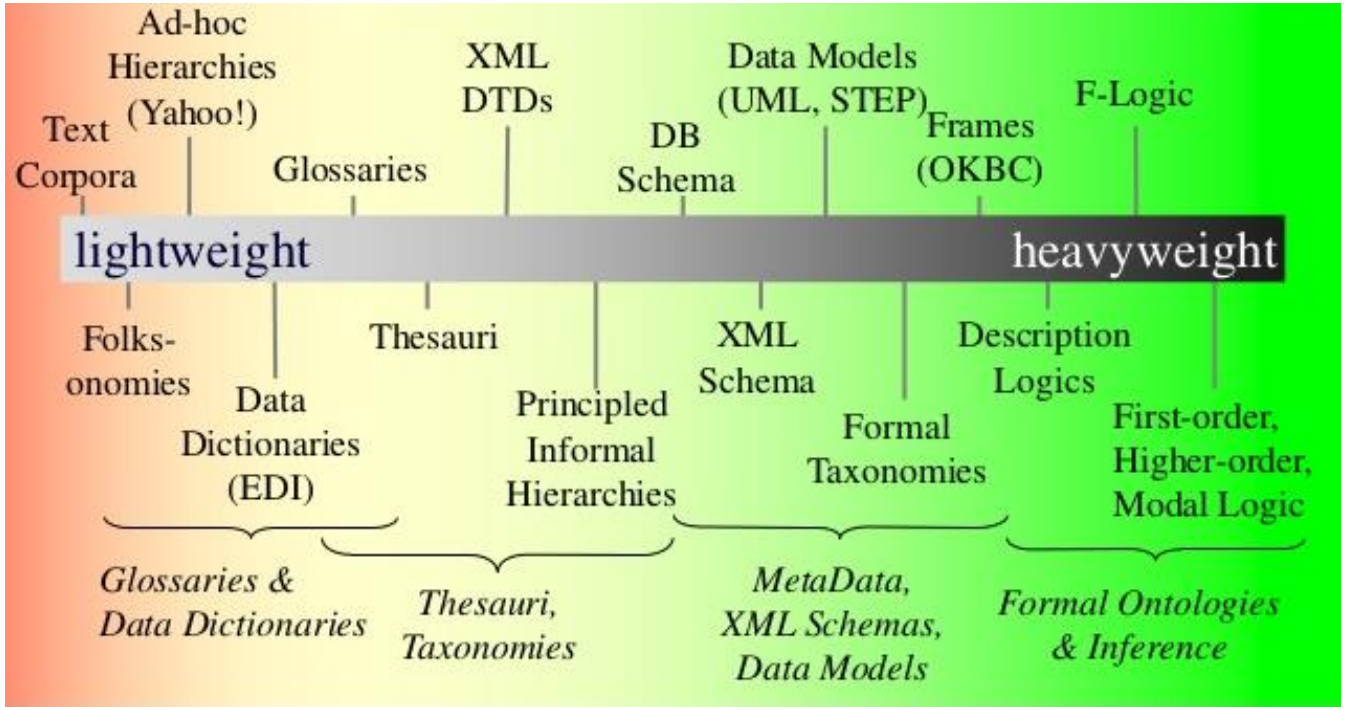


Attribute
(property, characteristics)

Types of ontology

Light weight ontologies

- glossaries, dictionaries
- thesauri
- taxonomies or conceptual hierarchies
- typically ***is-a*** relations



Fernando Silva Parreiras. *Model-Driven Software Development with Semantic Web Technologies.* http://www.slideshare.net/fparreiras/modeldriven-software-development-with-semantic-web-technologies

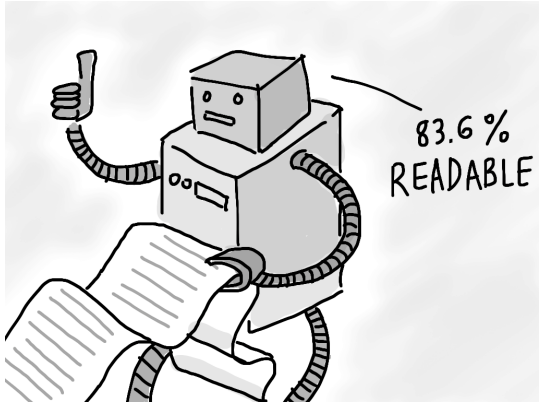
Heavy weight ontologies

- data models
- definition of concepts with axioms (i.e. established statements)
- logical formalisation and inference

Linking data in Semantic Web

- In Semantic Web, ontologies are **collections of statements** (written in e.g. Resource Description Framework *RDF*) that define the **relations between concepts** and specify **logical rules for reasoning** about them
- a common framework that allows data to be **shared and reused** across application, enterprise and community boundaries
- enables machines to “understand” and respond to complex human requests based on their meaning
- the relevant information sources need to be semantically structured

http://www.w3.org/standards/semanticweb/data

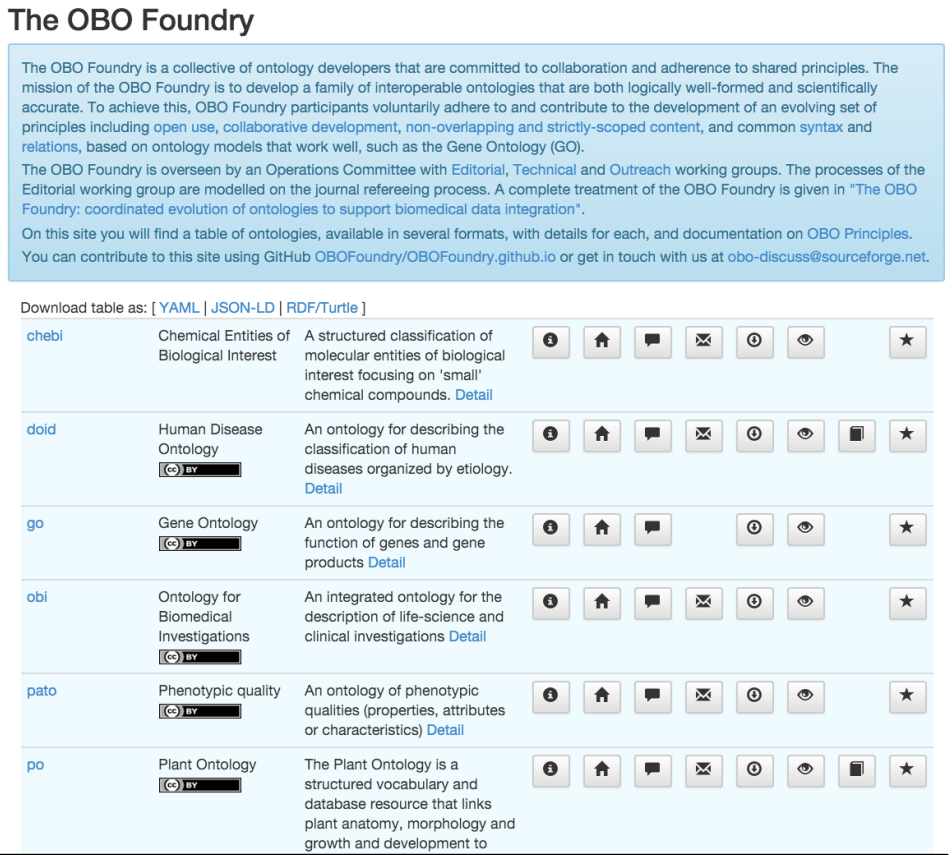


Biology is rapidly changing from a descriptive to a data-driven discipline in which the discovery of novel findings depends on the comparison and integration of massive data sets. As a consequence, **ontologies—systematic descriptions of specific biological attributes**—are becoming more and more important for describing the existing biological knowledge.

Jensen LJ & Bork P (2010) Ontologies in Quantitative Biology: A Basis for Comparison, Integration, and Discovery. *PLoS Biology* 8(5): e1000374.

Development of biological ontologies

- The OBO Foundry www.obofoundry.org
- Collective of ontology developers
- Adoption of a growing set of principles specifying best-practices in ontology development
- To foster **interoperability** of ontologies
- Projects include Human Disease Ontology, Gene Ontology etc.



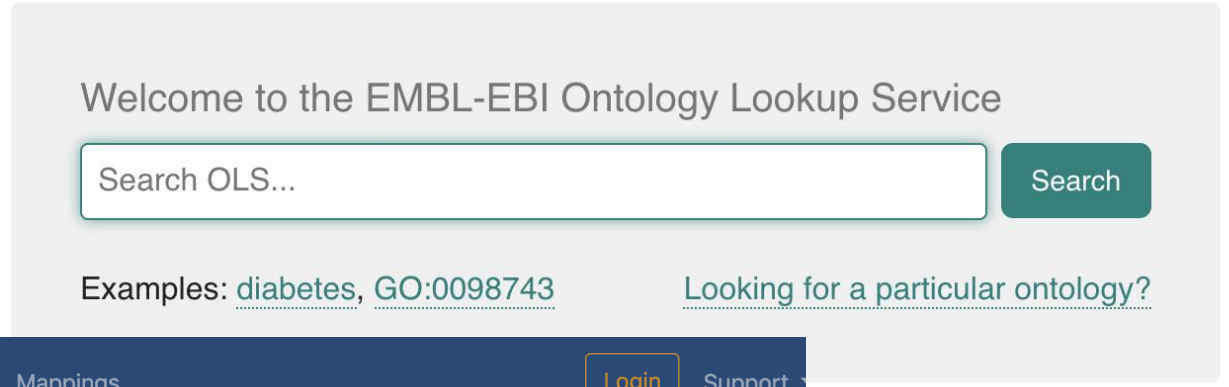
What does OBO stand for?
The Open Biological and Biomedical Ontologies Foundry (formerly known *as The Open Biomedical Foundry*, which was previously known as *The Open Biological Foundry*)

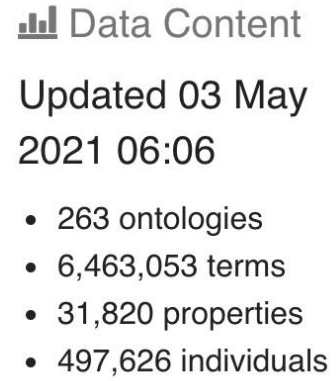
Smith B *et al.* (2007) *Nature Biotechnology* 25: 1251-1255.

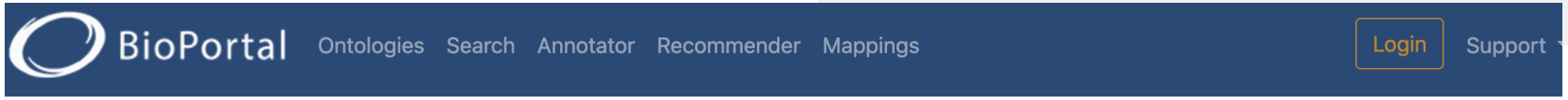
Resources of biological ontologies

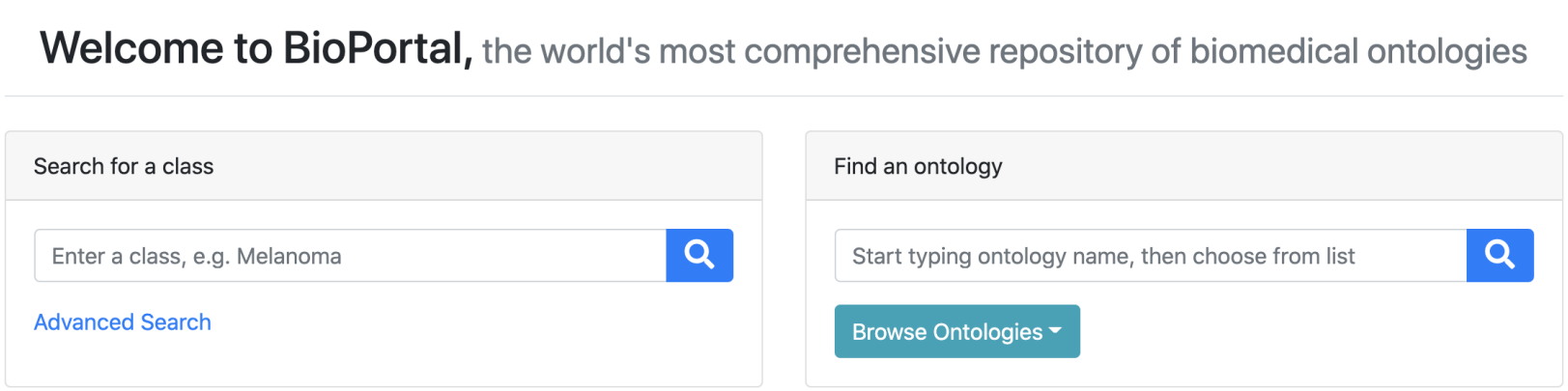
EBI Ontology Lookup Service
https://www.ebi.ac.uk/ols
- Ontologies from OBO Foundry

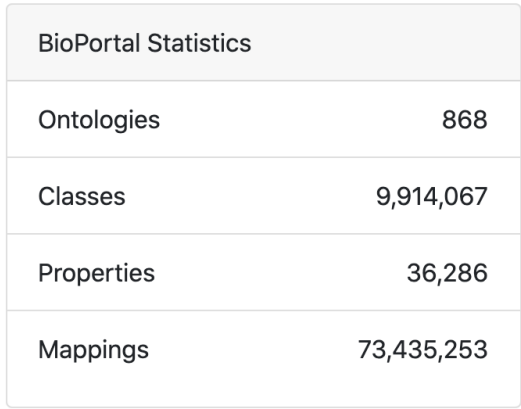
NCBO BioPortal
bioportal.bioontology.org
- open repository; user can add notes, review & map







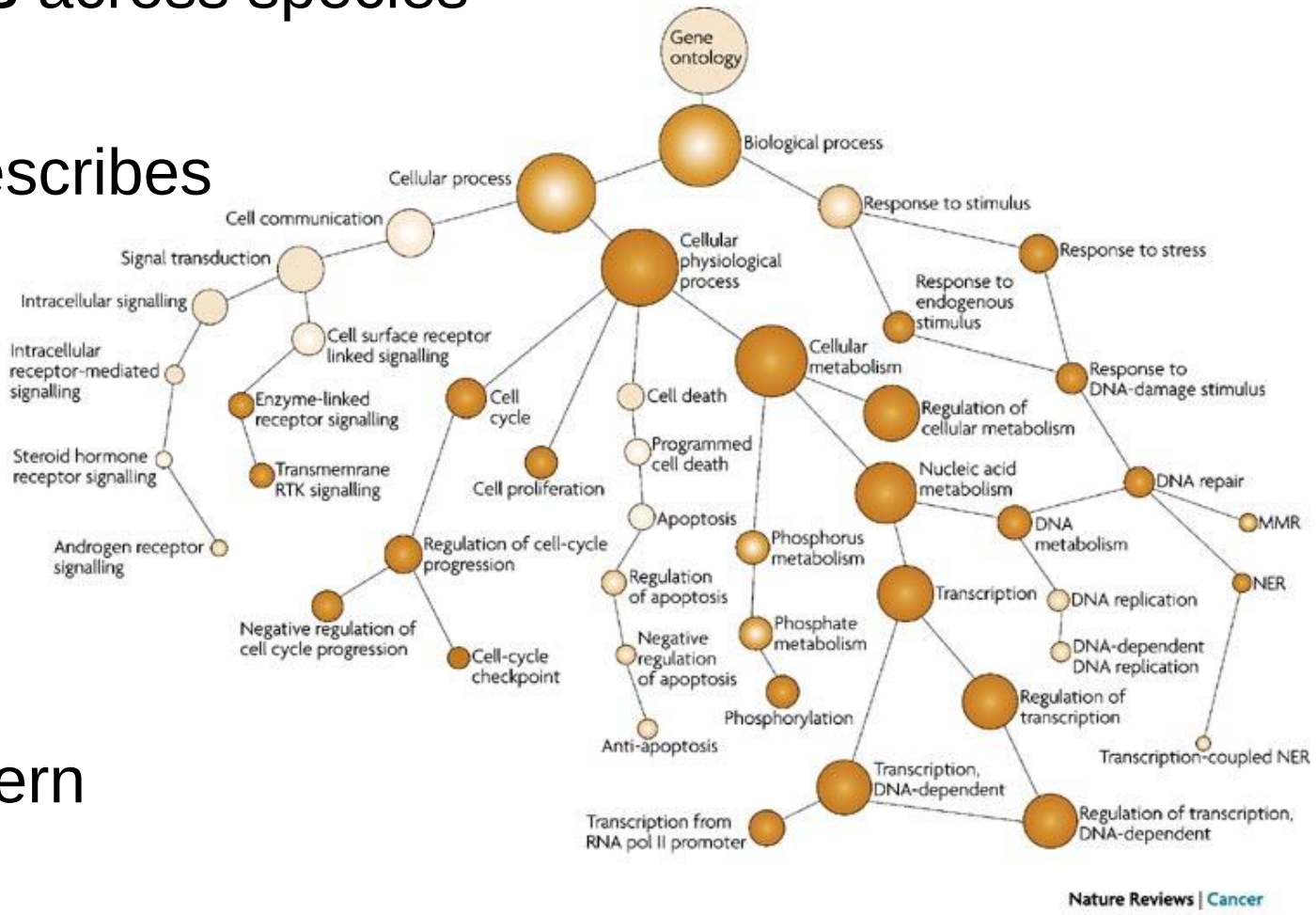




Gene Ontology

http://geneontology.org/

- Aims to standardise the representation of **gene and gene product attributes** across species and databases
- Controlled vocabulary that describes characteristics of gene products, and the associated annotation data
- Terms organised **resembling hierarchy**
- Widely used ontology in modern biological research



"The goal of the **Gene Ontology Consortium** is to produce **a dynamic, controlled vocabulary** that can be applied to all eukaryotes even as knowledge of gene and protein roles in cells is **accumulating and changing**."

Ashburner *et al.* (2000) *Nature Genetics*, 25: 25-29

Gene Ontology terms

http://geneontology.org/stats.html

Statistics for release 2021-02

Organised in **three** structured, **species-independent** ontologies that describe gene products based on their association to aspects of:

biological process | **cellular component** | **molecular function** ← Level 1 terms

Ontology

Property	Value
Valid terms	44085 (Δ = -6)
Obsoleted terms	3125 (Δ = 17)
Merged terms	2252 (Δ = 12)
Biological process terms	28748
Molecular function terms	11153
Cellular component terms	4184

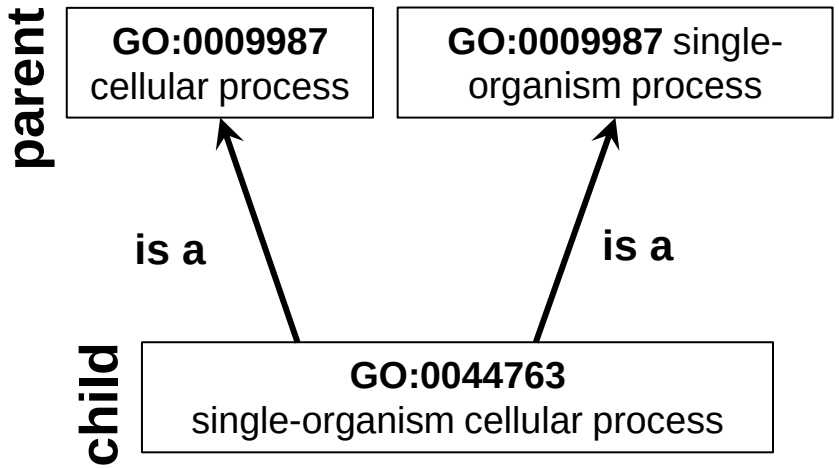
These ontologies resemble an **hierarchy**:

- **child terms** – more specialised; **parent terms** – less specialised
- a term may have **more than one** parent term (unlike an hierarchy)
- terms may be connected to parent terms via **different relations**

Example:

```
id: GO:0044763
name: single-organism cellular process
namespace: biological_process
def: "Any process that is carried out at the cellular
level, occurring within a single organism."
is_a: GO:0009987 ! cellular process
is_a: GO:0044699 ! single-organism process
```

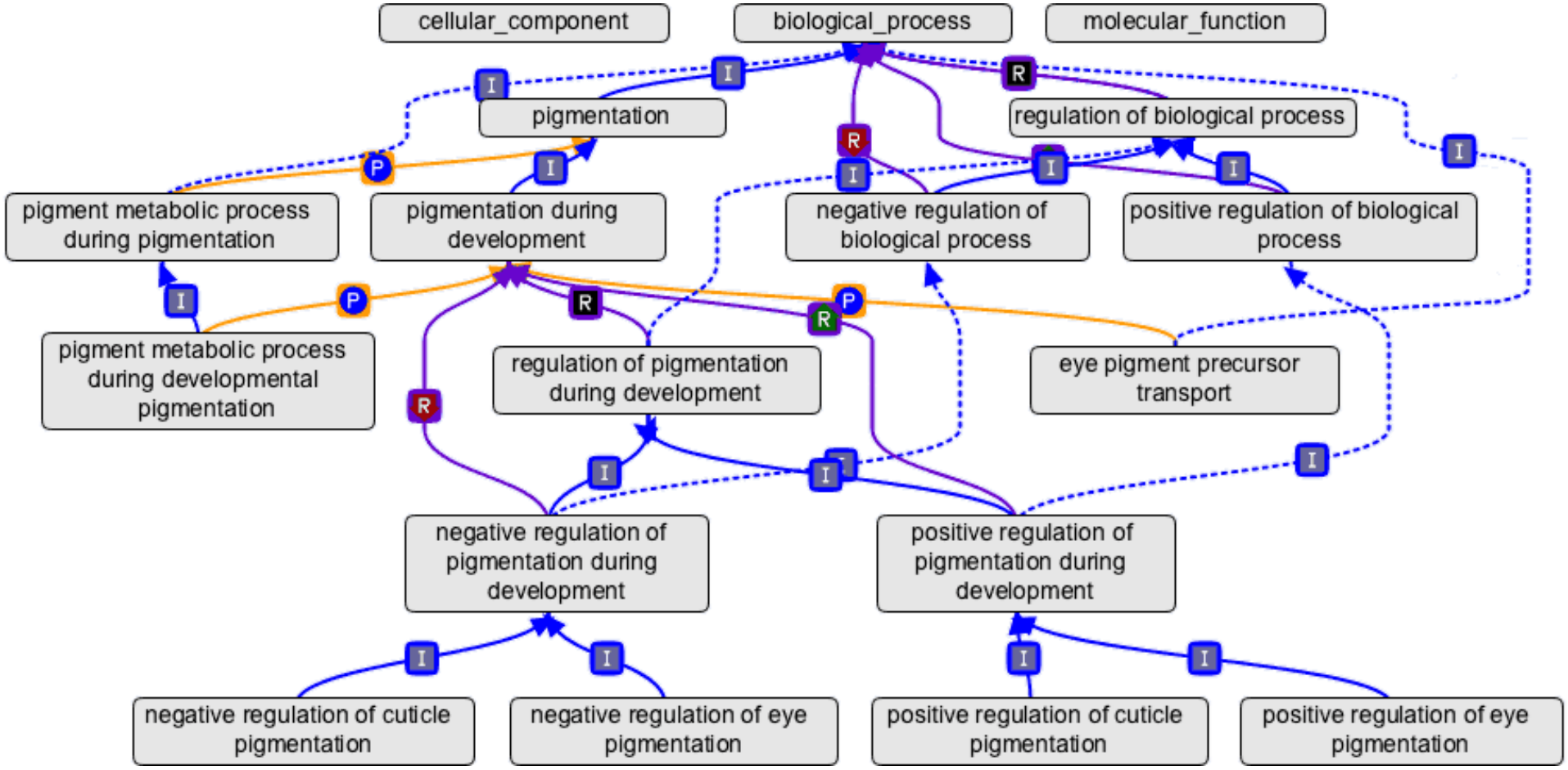
OBO format



Gene Ontology graph

Structure of GO can be described in terms of a graph:

- **node**: a GO term; **edge** (arc): the relationship between the two terms (nodes)

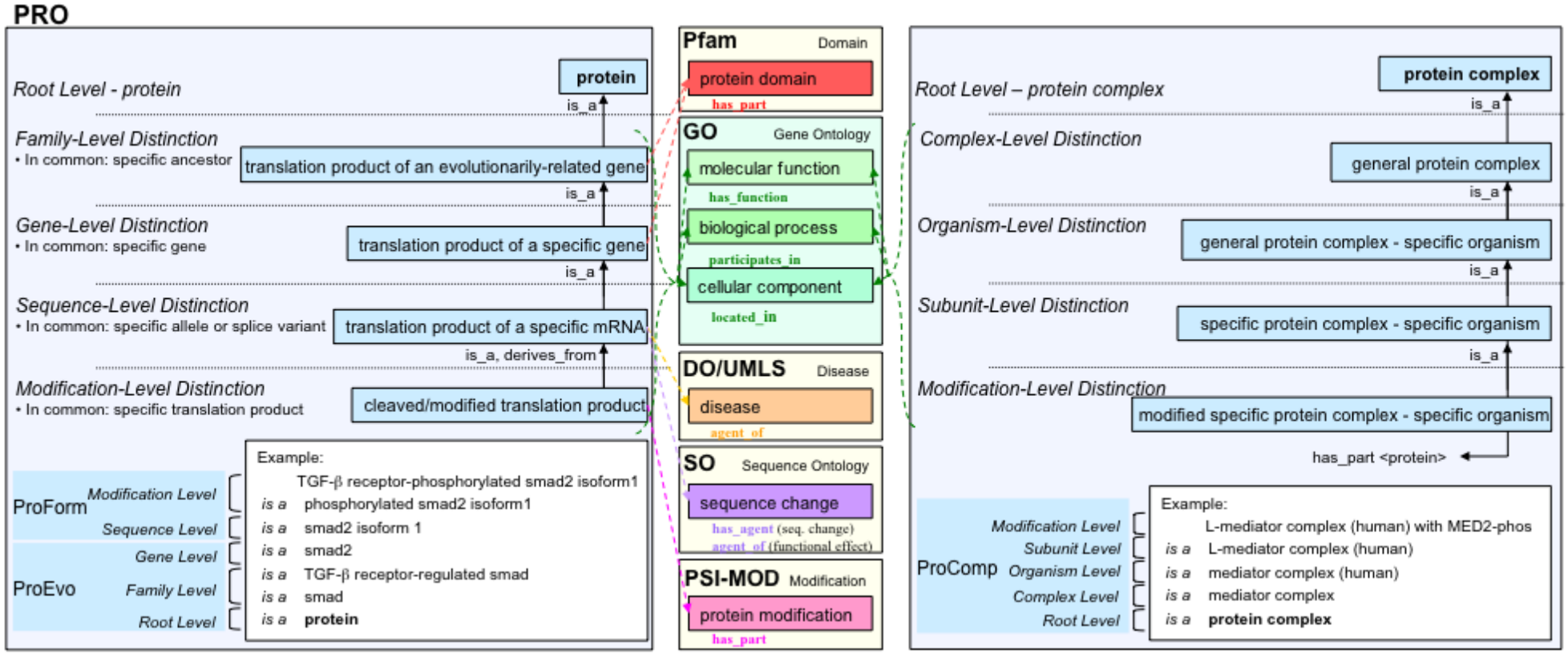


Relations **I: is a** **P: part of**
R: regulates (one process directly affects the manifestation of another process or quality)

Protein Ontology

PRO pir.georgetown.edu/pro/

- ontological representation of protein-related entities
- three sub-ontologies:
  - ProEvo (based on evolutionary relatedness)
  - ProForm (protein forms produced from a given gene locus); and
  - ProComp (protein-containing complexes)

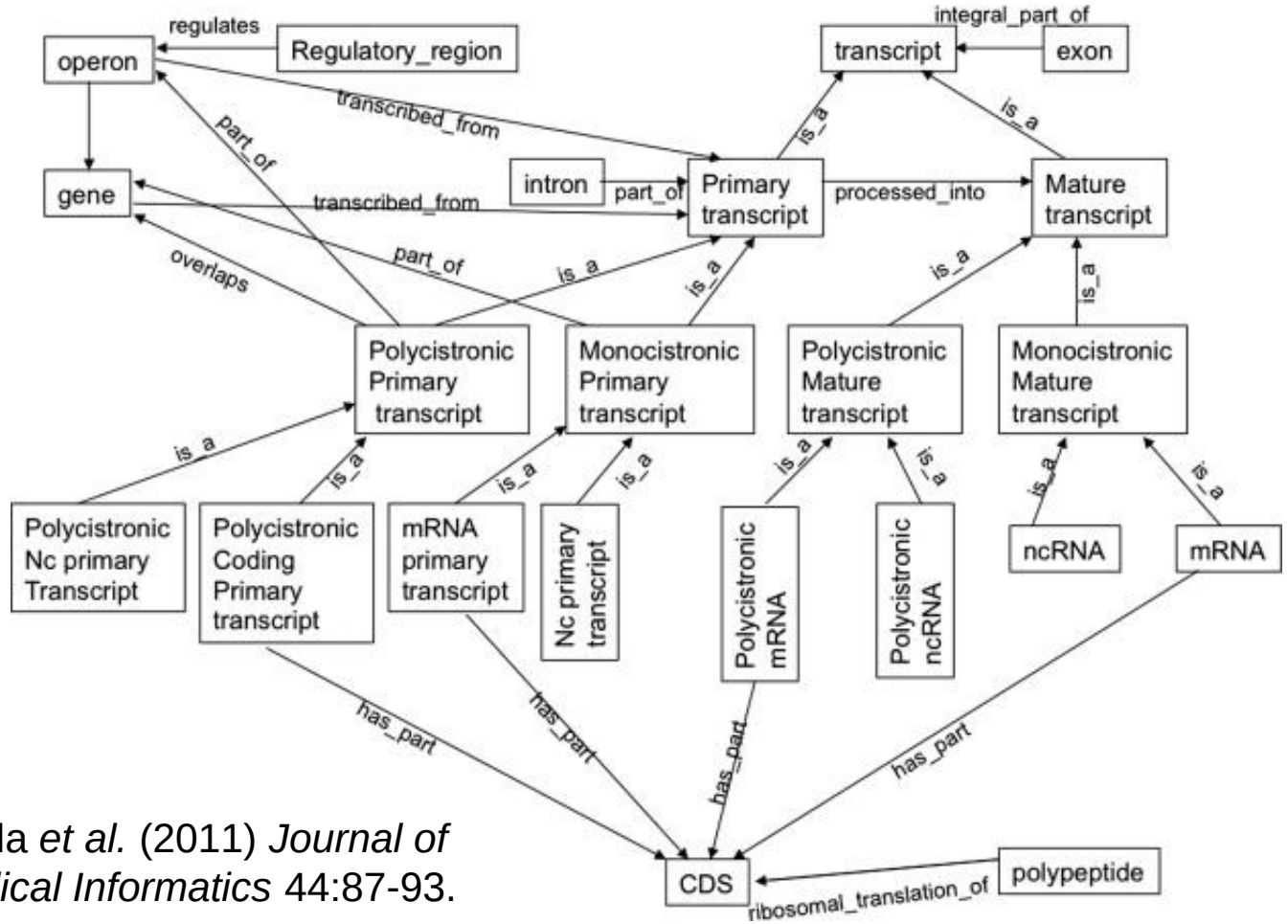


Sequence Ontology

SO: www.sequenceontology.org

- describes features and attributes of biological sequences, e.g. as defined by their disposition to be involved in a biological process, e.g. **binding_site** and **exon**

- describes primary annotations of nucleic acid sequence, and of mutations
- a structured SO within databases allows for query for e.g. *genes whose transcripts are edited, or trans-spliced, or are bound by a particular protein*.

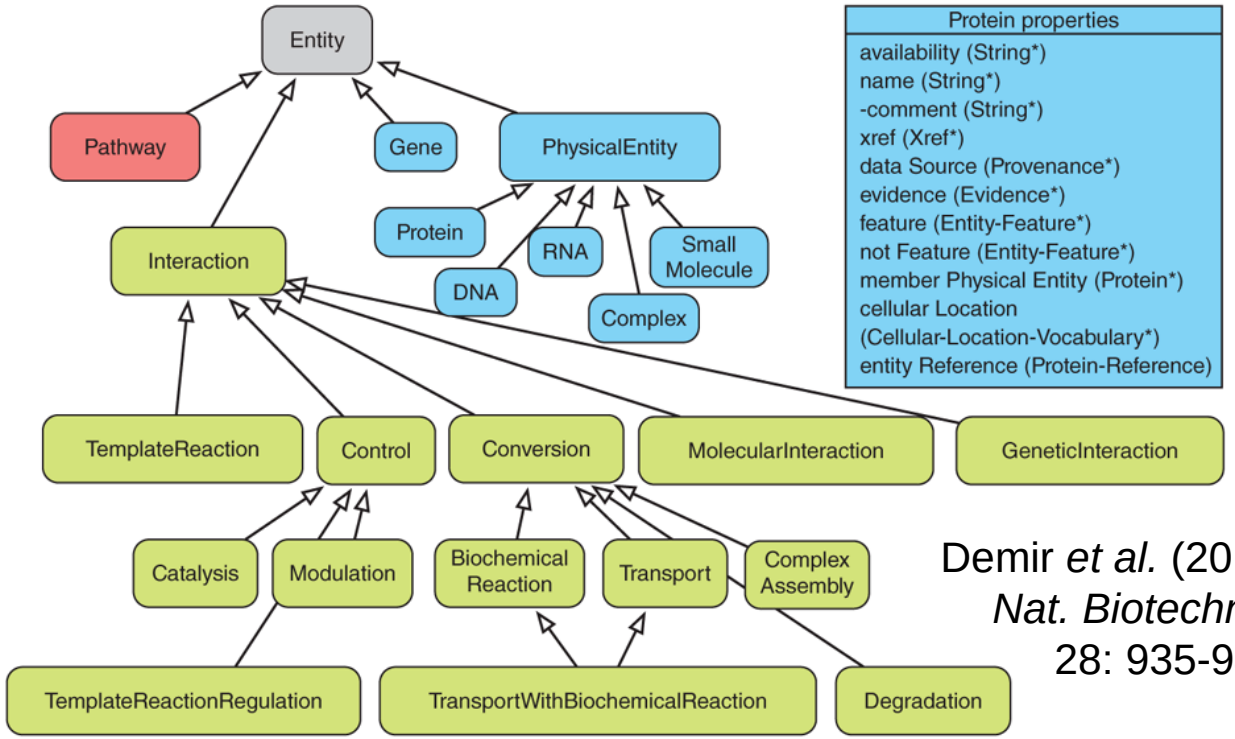


Mungalla *et al.* (2011) *Journal of Biomedical Informatics* 44:87-93.

Pathway Ontology

BioPax: www.biopax.org

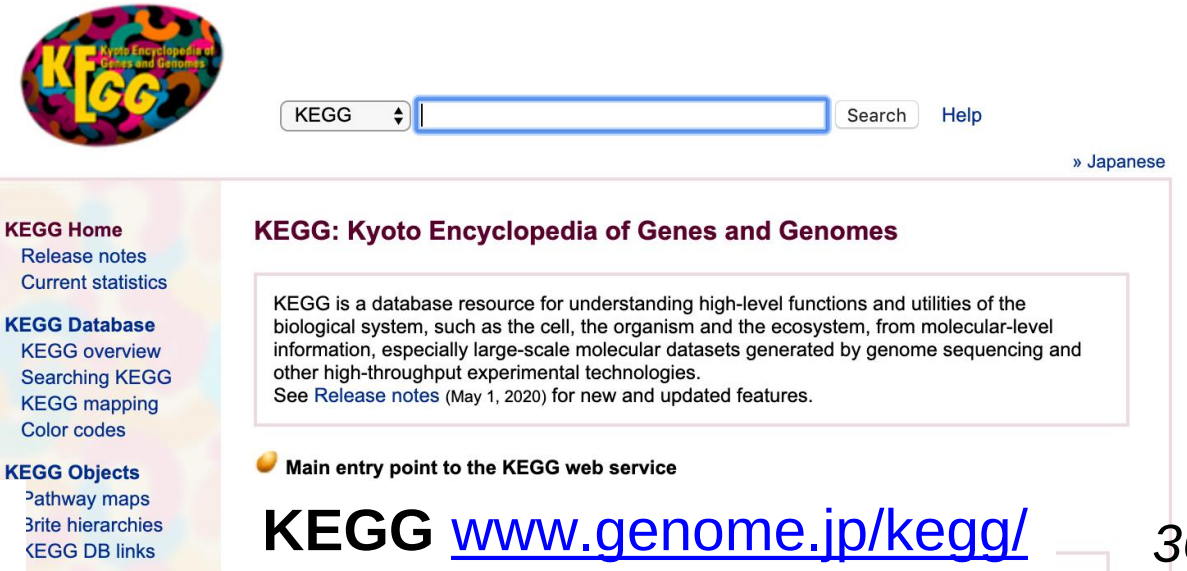
- formalisation of biochemical pathways; enables integration, exchange, visualisation and analysis of biological and signalling pathways, gene regulations, genetic interactions
- iterative development with increasing levels of biological knowledge modelled



Demir *et al.* (2010) *Nat. Biotechnol.* 28: 935-942.

Examples of databases:

Reactome www.reactome.org



KEGG www.genome.jp/kegg/

Other examples of biological ontologies

- Species-specific
  - *C. elegans* phenotype (wbphenotype)
- Chemical entities
  - Chemical Entities of Biological Interest (chebi)
- Molecular interactions
  - Protein modification (PSI-MOD), molecular interactions (PSI-MI)
- Investigations/experiments
  - Ontology for Biomedical Investigations (obi)
- Biomedical literature
  - Medical Subject Headings (MeSH)
- Many more …

Most are listed in www.obofoundry.org

https://www.ebi.ac.uk/chebi/

www.nlm.nih.gov/mesh/

Ontobee: http://www.ontobee.org/

ontology data sharing, visualization, query, integration, and analysis

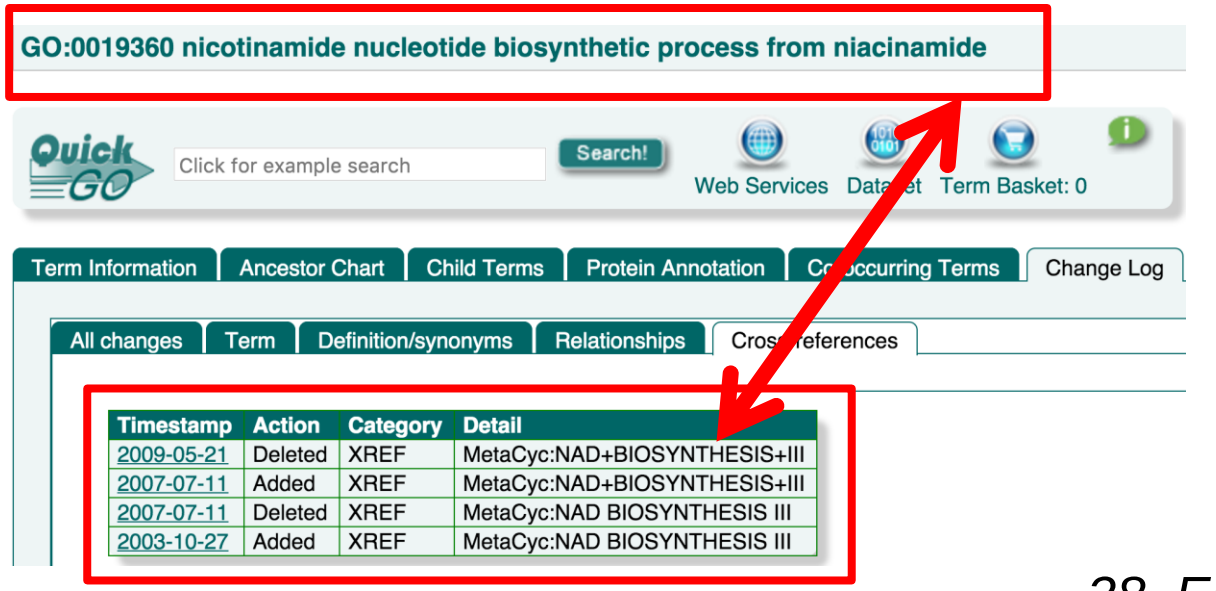
Biological ontologies: Issues and challenges

- Ontologies for complex entities e.g. **genotypes and phenotypes**; some on-going projects include:
  - Genotype ontology (GENO) to characterise genetic variation
  - Human Phenotype Ontology (HPO) Project: to phenotypic abnormalities encountered in human disease

https://hpo.jax.org/app/

- Terms represented in **several**, possibly **overlapping**, ontologies - may cause errors in data cross-linking

- **Funding** support for maintenance



Database 2

Searching for sequences in databases

Cheong Xin Chan (CX)

c.chan1@uq.edu.au

Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics 1 – Introduction

Outline

- **Basic concepts of sequence searching**
  - The general approach and concept of sub-sequences (*k*-tuples or *k*-mers)
- **FastA**
  - Basic principles
  - Hashing-and-chaining algorithm
- **Basic Local Alignment Search Tool (BLAST)**
  - Differences between FastA and BLAST
  - Basic parameters of BLAST
  - BLAST algorithm using Finite State Machine

The basic concepts

- to search for **similar** sequences (in a database) to a query, **alignment** is necessary
- Needleman-Wunsch or Smith-Waterman algorithm is too slow for this purpose; faster approach was developed: **FastA** and **BLAST**
- assumption: a good local alignment should have some identical subsequences (i.e. **exact matches** of sub-sequences)
- these sub-sequences at a fixed length *k*, are referred to as ***k*-tuples** or ***k*-mers**
- usually, smaller *k* (e.g. 2–3) is used for protein sequences; larger *k* (e.g. 3–6) for DNA sequences

Example

Protein 2-tuples (*k* = 2) → **AN**, **AR**, …

DNA 4-tuples (*k* = 4) → **TAAA**, **TAAC**, …

General approach for sequence searching

1. Pre-process **query** string, e.g. generate short sub-sequences to search for
2. Quickly **align sub-sequences** with sequences in the database, keeping only high-scoring sub-alignments
3. Attempt to **join sub-alignments**, creating tentative scores
4. Perform a **thorough alignment** on high-scoring sequences

Sub-sequences at defined length ***k*** are known variously as ***k*-tuples**, ***k*-mers**, **words**, or ***n*-grams** (of length *n*)

FastA: searching similar sequences

- developed by Lipman & Pearson (1985)
- first described as FastP (P for protein); FastA (A for all) work on both nucleotide and protein sequences
- **FASTA** format – the most common text-based representation of biological sequences

David L. Lipman

William R. Pearson

FastA Server: http://fasta.bioch.virginia.edu/fasta_www2/

Key steps:

1. observe the pattern of word hits
2. identify word-to-word matches of a given length
3. mark potential matches
4. perform an optimised search using a Smith-Waterman type of algorithm

FastA: searching using hashing and chaining

Hashing

- an efficient approach for data storage and retrieval, common in computing
- in FastA, each distinct data entry (e.g. *k*-tuples) is assigned a **unique** integer (this saves storage space), i.e. the **index** (or *key*)
- relevant data (the *values*), i.e. position(s) at which the corresponding *k*-tuple is found on the query sequence, are stored in relation to each **index**
- data are structured in a *key-value* relationship

FastA: searching using hashing and chaining

Hashing: assigning indices

At k = 3 for nucleotide sequences:

- there are 4^3 = **64** possible 3-tuples
- let number $\mathbf{e(N)}$ be a distinct value for each nucleotide $\mathbf{N}$: $\mathbf{e(A) = 0}$, $\mathbf{e(C) = 1}$, $\mathbf{e(G) = 2}$ and $\mathbf{e(T) = 3}$
- a 3-tuple, represented as $x_i\, x_{i+1}\, x_{i+2}$, is assigned a value C_i:

$$C_i = e(x_i)4^2 + e(x_{i+1})4^1 + e(x_{i+2})4^0$$

$$C_i(\mathbf{AAA}) = e(\mathbf{A})4^2 + e(\mathbf{A})4^1 + e(\mathbf{A})4^0 = 0\times4^2 + 0\times4^1 + 0\times4^0 = \mathbf{0}$$

$$C_i(\mathbf{CAA}) = e(\mathbf{C})4^2 + e(\mathbf{A})4^1 + e(\mathbf{A})4^0 = 1\times4^2 + 0\times4^1 + 0\times4^0 = \mathbf{16}$$

Example

- the C_i is the **index** representation of the 3-tuple

FastA: searching using hashing and chaining

Chaining: creating a look-up table

record the position(s) on a sequence at which the *k*-tuples occurred (the *values*), and assign to the corresponding **index** (the *key*)

Example

consider **TAAAACTCTAAC** (at *k* = 3):

	Index (*key*)	**Position(s)** (*values*)
AAA →	0	2, 3
AAC →	1	4, 10
AAG →	2	–
AAT →	3	–
…	…	…
TTT →	63	–

FastA: searching using hashing and chaining

Chaining:

consider **TAAAACTCTAAC** (at $k = 3$):

Example

Position number	1	2	3	4	5	6	7	8	…
Nucleotide	T	A	A	A	A	C	T	C	…

Index of 3-tuple (C_i)	AAA	AAC	AAG	AAT	ACA	ACC	ACG	ACT	…	TAA	…
First position where the 3-tuple is found on the sequence	2	4	0	0	0	0	0	5	…	1	…

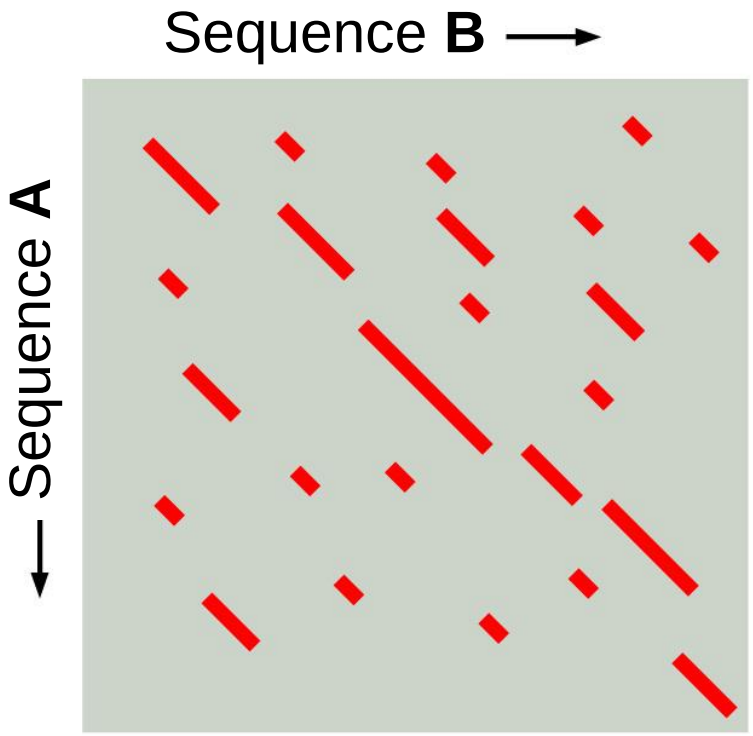
Position number	1	2	3	4	5	6	7	…
Next position the same 3-tuple is found	9	3	0	10	0	0	0	…

AAA found at position **2** then at **3** …. ; **AAC** found at position **4** then at **10** ... etc.

FastA: searching using hashing and chaining

Step 1: *observe the pattern of word hits*

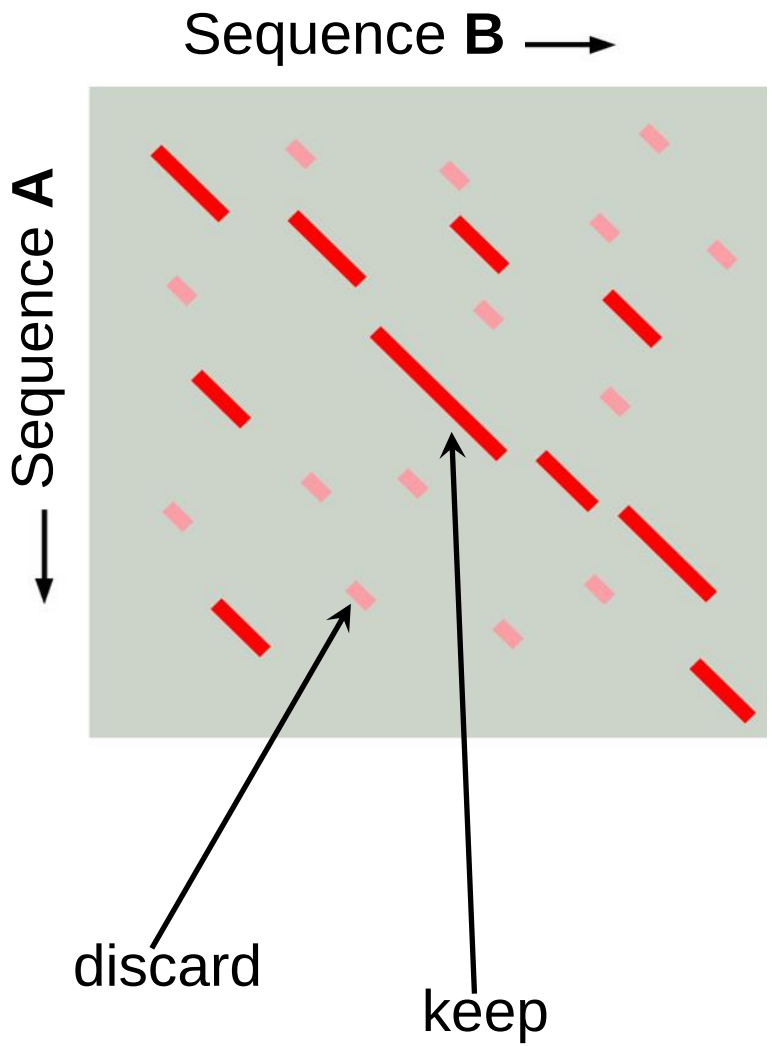
- identify perfect matches between sequence **A** and sequence **B**, using *k*-tuples *via* **hashing and chaining**
- look for high-density local regions between the two sequences; these are locally aligned regions



FastA: searching using hashing and chaining

Step 2: identify word-to-word matches of a given length

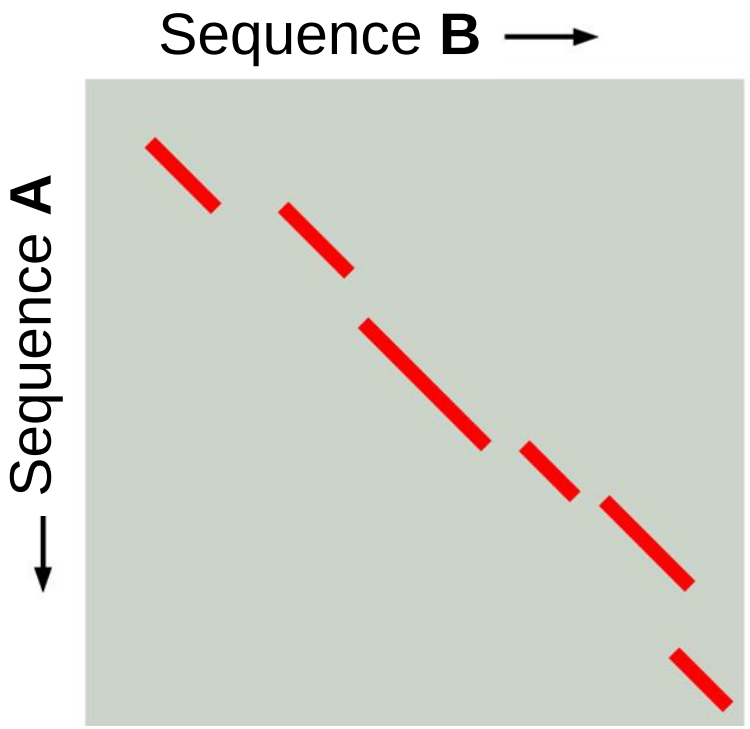
- re-score the aligned regions using a substitution scoring matrix
- keep only those contributing to the highest score, e.g. the top 10 aligned regions; discard the others



FastA: searching using hashing and chaining

Step 3: *mark potential matches*

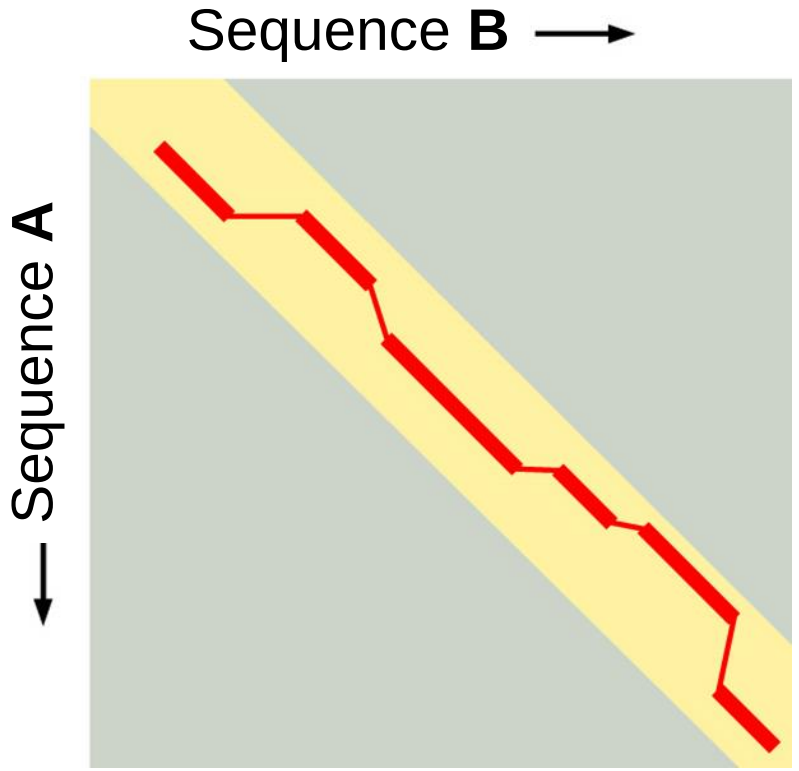
- initial regions with scores greater than a cut-off value are *joined* into an approximate alignment with gaps
- **initial similarity score** (“init***n***”) is calculated from scores of individual alignments with joining penalties
- this score is used for **preliminary ranking** of database sequences
- **init1** is the score of the single best initial region (i.e top of the rank)



FastA: searching using hashing and chaining

Step 4: *local alignment using SW algorithm*

- for suitably high-scoring database sequences, local alignment is performed between the two sequences using a Smith-Waterman algorithm
- The final score, “opt” is used for final sequence ranking; significance is calculated



BLAST: Basic Local Alignment Search Tool

- developed by Altschul et al. (1990)
- search for **High-scoring Segment Pairs** (HSPs) contained in a **statistically significant** alignment
- most commonly used tool for sequence searching in major databases: http://blast.ncbi.nlm.nih.gov/Blast.cgi
- similar approach to FastA but with **two** key differences:

FastA	**BLAST**
uses identical *k*-tuples (**exact matches**)	uses *k*-mers in the target that score above a **threshold *T*** (allowing for **non-exact matches** for protein sequences)
algorithm based on **hashing and chaining**	algorithm based on **finite state machine** (FSM), also known as **finite state automata** (FSA)

BLAST: the key parameters

Word size, i.e. *k* in *k*-mer

- typically **3** (2-4) for protein sequences, **11** for nucleotide sequences

Threshold *T* (typically an integer between 11 and 19):

- only *k*-mer (exact or inexact) matches that score **equal to or greater than *T*** (a neighbourhood score threshold) are used for seeding

Scoring (substitution) matrix

- typically BLOSUM62; the optimal scoring matrix depends on expected sequence similarity

Drop-off *X*

- the amount of drop in score that will stop the sequence extension, typically 20

BLAST algorithm

Phase 1: *remove low-complexity region or repeats in the query sequence*

```
LLQQQQAVMLQQQQLQEFYKKQQEQLHLQLLQQQQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQHPGKQAKEQQQQQQLAAQQLVFQQ
```

Example

Low-complexity region in a sequence composes of very little variation (i.e. fewer distinct residues)

- they might yield high alignment scores but not due to biologically significance, e.g. between two non-homologous sequences (simply by chance)
- these regions in the sequence database and in the query are commonly masked before a search

BLAST algorithm

Phase 2: *identify k-mers in query with scores above threshold T for seeding*

Finite state machine (FSM)
a computational algorithm used to model a variety of phenomena that are described by a succession of states that control the behavior of a system

- given the current state and an input, the next state of the FSM can be **deterministically determined** (not probabilistic)
- given a list of *k*-mers in the query (input), the FSM can quickly produce all possible matches with scores exceeding a **threshold *T***; these *k*-mers are used for seeding (to start the alignment process)

BLAST algorithm

Phase 2: *identify k-mers in query with scores above threshold T for seeding*

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	−1	4																		
T	−1	1	5																	
P	−3	−1	−1	7																
A	0	1	0	−1	4															
G	−3	0	−2	−2	0	6														
N	−3	1	0	−2	−2	0	6													
D	−3	0	−1	−1	−2	−1	1	6												
E	−4	0	−1	−1	−1	−2	0	2	5											
Q	−3	0	−1	−1	−1	−2	0	0	2	5										
H	−3	−1	−2	−2	−2	−2	1	−1	0	0	8									
R	−3	−1	−1	−2	−1	−2	0	−2	0	1	0	5								
K	−3	0	−1	−1	−1	−2	0	−1	1	1	−1	2	5							
M	−1	−1	−1	−2	−1	−3	−2	−3	−2	0	−2	−1	−1	5						
I	−1	−2	−1	−3	−1	−4	−3	−3	−3	−3	−3	−3	−3	1	4					
L	−1	−2	−1	−3	−1	−4	−3	−4	−3	−2	−3	−2	−2	2	2	4				
V	−1	−2	0	−2	0	−3	−3	−3	−2	−2	−3	3	2	1	3	1	4			
F	−2	−2	−2	−4	−2	−3	−3	−3	−3	−3	−1	−3	−3	0	0	0	−1	6		
Y	−2	−2	−2	−3	−2	−3	−2	−3	−2	−1	2	−2	−2	−1	−1	−1	−1	3	7	
W	−2	−3	−2	−4	−3	−2	−4	−4	−3	−2	−2	−3	−3	−1	−3	−2	−3	1	2	11

query

Consider a 3-mer **CHH**, and ***T* = 19**

Based on BLOSUM62 matrix, each of the following transitions has a score ≥ 19:

- **CHH→CHH**: 9 + 8 + 8 = **25**
- **CHH→CHY**: 9 + 8 + 2 = **19**
- **CHH→CYH**: 9 + 2 + 8 = **19**

Example

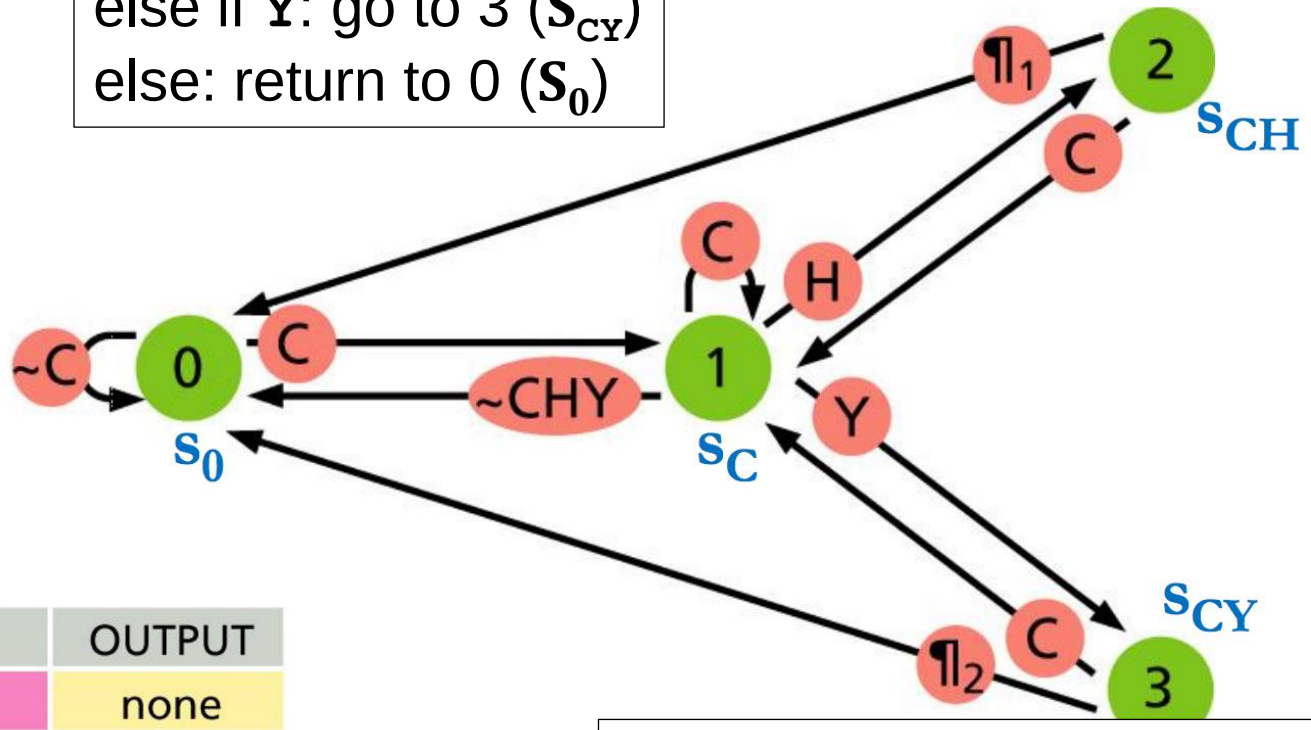
FSM

Example

State 0 (S_0):
if **C**: go to 1 (S_C)
else: stay here

State 1 (S_C):
if **C**: stay here
else if **H**: go to 2 (S_{CH})
else if **Y**: go to 3 (S_{CY})
else: return to 0 (S_0)

State 2 (S_{CH}):
if **H**: output **CHH** & return to 0 (S_0)
else if **Y**: output **CHY**, return to 0 (S_0)
else if **C**: return to 1 (S_C)
else: return to 0 (S_0)



anything but **CHY**

anything but **CH**

	INPUT	OUTPUT
¶1:	~CHY	none
	H	CHH
	Y	CHY
¶2:	~CH	none
	H	CYH

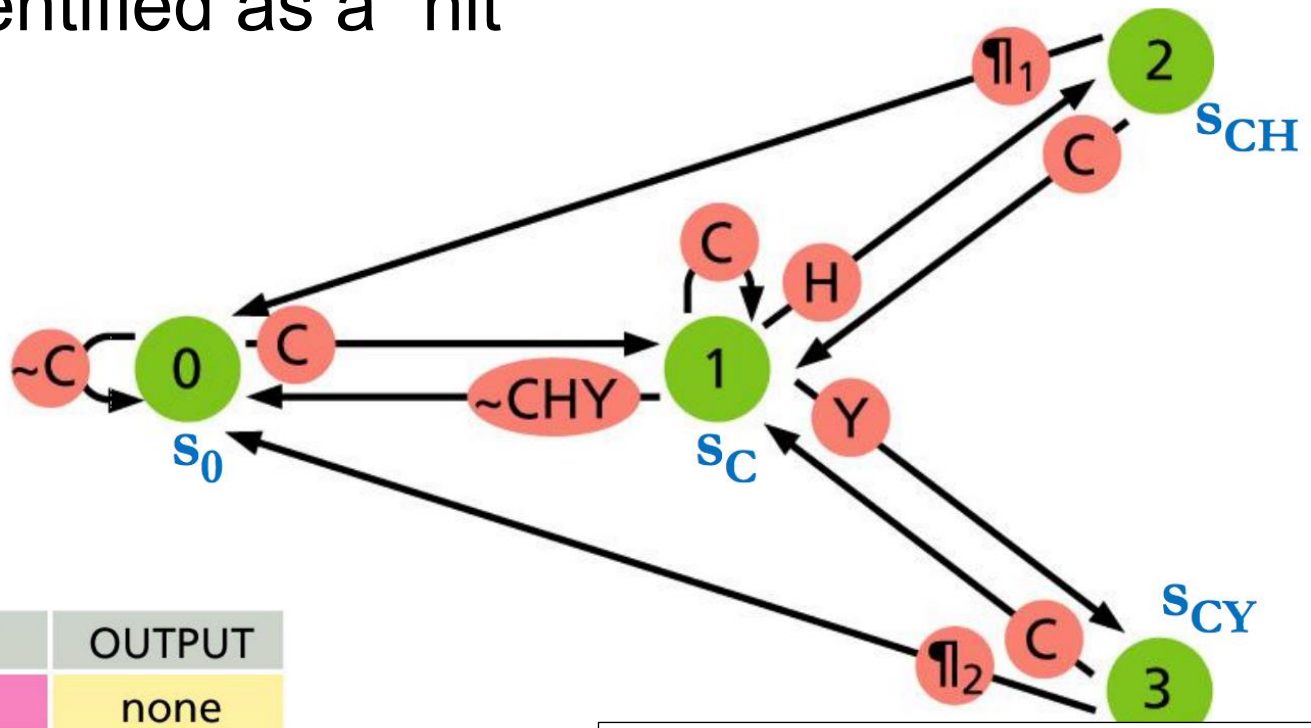
State 3 (S_{CY}):
if **H**: output **CYH** & return to 0 (S_0)
else if **C**: return to 1 (S_C)
else: return to 0 (S_0)

FSM

Example

e.g. a (subject) sequence from the database

If the input is **CHCYHC**, the states visited are **0–1–2–1–3–0–1**, with CYH identified as a “hit”



	INPUT	OUTPUT
¶1:	~CHY	none
	H	CHH
	Y	CHY
¶2:	~CH	none
	H	CYH

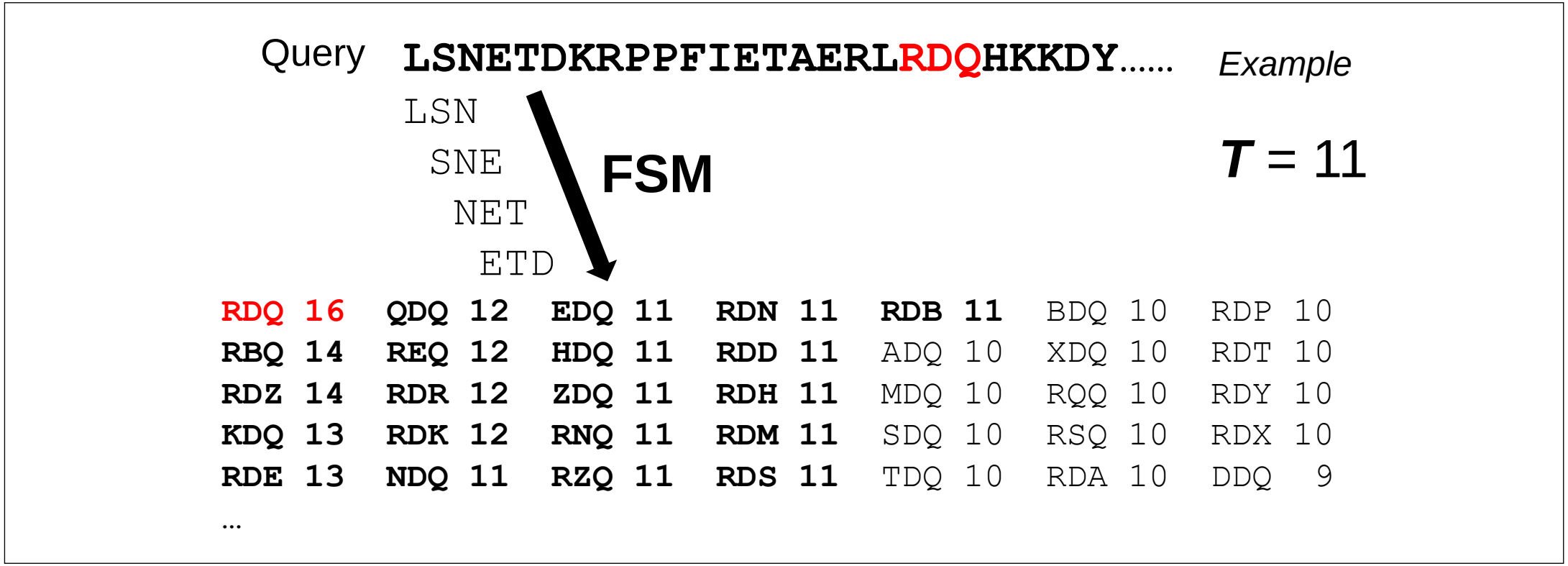
anything but **CHY**

anything but **CH**

State 3 (S_{CY}):
if **H**: output **CYH** & return to 0 (S_0)
else if **C**: return to 1 (S_C)
else: return to 0 (S_0)

BLAST algorithm

Phase 2: *identify k-mers in query with scores above threshold T for seeding*



This list represents *k*-mers (in any target/database sequence) that would give a score of ***T*** or higher when aligned with the query sequence; they are used for **seeding**

BLAST algorithm

Phase 3: *identify matches of these k-mers in each database sequence*

For each target sequence in the database:

- scan for hits with the compiled list of *k*-mers in Phase 2
- extend the hits to form **high-scoring segment pairs (HSPs)**
- Extension proceeds in **both** directions from the *k*-mer until the score drops by more than ***X*** relative to the current best score
- identify the highest scoring pair, i.e. maximal segment pair (MSP); if needed, combine two or more HSPs into a longer alignment
- final alignment score, ***S***

BLAST algorithm

Query: GSVEDTTGSQSLAALLNKCKT**PQG**QRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL

PQG 18
PEG 15
PRG 14
PKG 14
PNG 13
PDG 13
PHG 13
PMG 13
PSG 13
PQA 12
PQN 12
etc.

T = 13

extension proceeds in both directions

```
Query:   325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
             +LA++L+   TP G R++ +W+  P+ D   + ER   + A
Sbjct:   290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
```

High-scoring segment pair (HSP)

Example

BLAST algorithm

Phase 4: *evaluate the statistical significance of the alignments*

Expect Value (*E*-value)

- the number of alignments with a score **equal to or greater than** the observed score, that would be expected by chance alone in searching a database of n sequences
- *E*-value = 1: we would expect to see 1 match with the observed score or higher simply by chance
- the smaller an *E*-value, the more “significant” the match is
- dependent on the size of the database