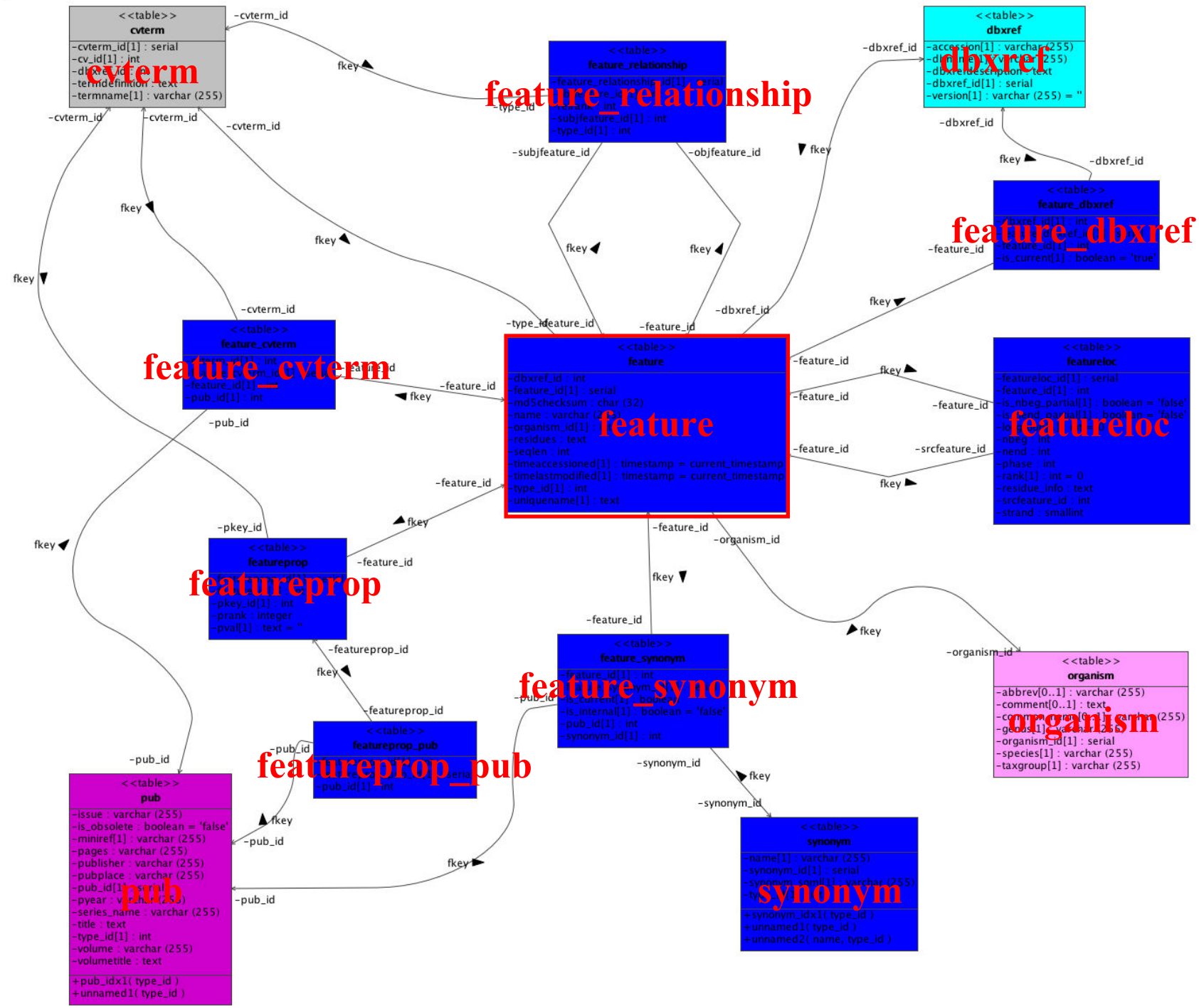
chado

Generic model organism database schema

presented at Cold Spring Harbor GMOD meeting, 5/03
by Stan Letovsky, sletovsky@aol.com

Chado modules

Genetics Module	Alleles, and relationships between alleles and phenotypes
Sequence Module	Anything related to biological sequences and annotation
Map Module	Any kind of localisation excluding sequence localisations
Expression Module	Transcription events, including space/time localisation. Also for protein expression
Companalysis Module	Adjunct to sequence module for in-silico analysis
CV Module	Controlled Vocabularies / Ontologies
Organism Module	Data related to taxonomy / species
Pub Module	Publication / Biblio / References
General Module	General / Core


cvterm
<<table>>
cvterm
-cvterm_id[1] : serial
-cv_id[1] : int
-dbxref_id : int
-termdefinition : text
-termname[1] : varchar (255)
feature_relationship
<<table>>
feature_relationship
-feature_relationship_id[1] : serial
-type_id : int
-rank : int
-subjfeature_id[1] : int
-type_id[1] : int
dbxref
<<table>>
dbxref
-accession[1] : varchar (255)
-dbname[1] : varchar (255)
-dbxrefdescription : text
-dbxref_id[1] : serial
-version[1] : varchar (255) = ''
feature_dbxref
<<table>>
feature_dbxref
-dbxref_id[1] : int
-feature_id[1] : int
-is_current[1] : boolean = 'true'
feature_cvterm
<<table>>
feature_cvterm
-cvterm_id[1] : int
-feature_id[1] : int
-pub_id[1] : int
feature
<<table>>
feature
-dbxref_id : int
-feature_id[1] : serial
-md5checksum : char (32)
-name : varchar (255)
-organism_id[1] : int
-residues : text
-seqlen : int
-timeaccessioned[1] : timestamp = current_timestamp
-timelastmodified[1] : timestamp = current_timestamp
-type_id[1] : int
-uniquename[1] : text
featureloc
<<table>>
featureloc
-featureloc_id[1] : serial
-feature_id[1] : int
-is_nbeg_partial[1] : boolean = 'false'
-is_nend_partial[1] : boolean = 'false'
-locgroup[1] : int = 0
-nbeg : int
-nend : int
-phase : int
-rank[1] : int = 0
-residue_info : text
-srcfeature_id : int
-strand : smallint
featureprop
<<table>>
featureprop
-pkey_id[1] : int
-prank : integer
-pval[1] : text = ''
feature_synonym
<<table>>
feature_synonym
-feature_id[1] : int
-synonym_id[1] : int
-is_current[1] : boolean = 'false'
-is_internal[1] : boolean = 'false'
-pub_id[1] : int
-synonym_id[1] : int
organism
<<table>>
organism
-abbrev[0..1] : varchar (255)
-comment[0..1] : text
-common_name[0..1] : varchar (255)
-genus[1] : varchar (255)
-organism_id[1] : serial
-species[1] : varchar (255)
-taxgroup[1] : varchar (255)
featureprop_pub
<<table>>
featureprop_pub
-featureprop_id[1] : int
-pub_id[1] : int
pub
<<table>>
pub
-issue : varchar (255)
-is_obsolete : boolean = 'false'
-miniref[1] : varchar (255)
-pages : varchar (255)
-publisher : varchar (255)
-pubplace : varchar (255)
-pub_id[1] : serial
-pyear : varchar (255)
-series_name : varchar (255)
-title : text
-type_id[1] : int
-volume : varchar (255)
-volumetitle : text
+pub_idx1(type_id)
+unnamed1(type_id)
synonym
<<table>>
synonym
-name[1] : varchar (255)
-synonym_id[1] : serial
-synonym_sgml[1] : varchar (255)
-type_id[1] : int
+synonym_idx1(type_id)
+unnamed1(type_id)
+unnamed2(name, type_id)
-cvterm_id
-dbxref_id
-subjfeature_id
-objfeature_id
-type_id
-feature_id
-pub_id
-pkey_id
-featureprop_id
-organism_id
-synonym_id
-srcfeature_id
fkey

Sequence Ontology

DAG-Edit version 1.320

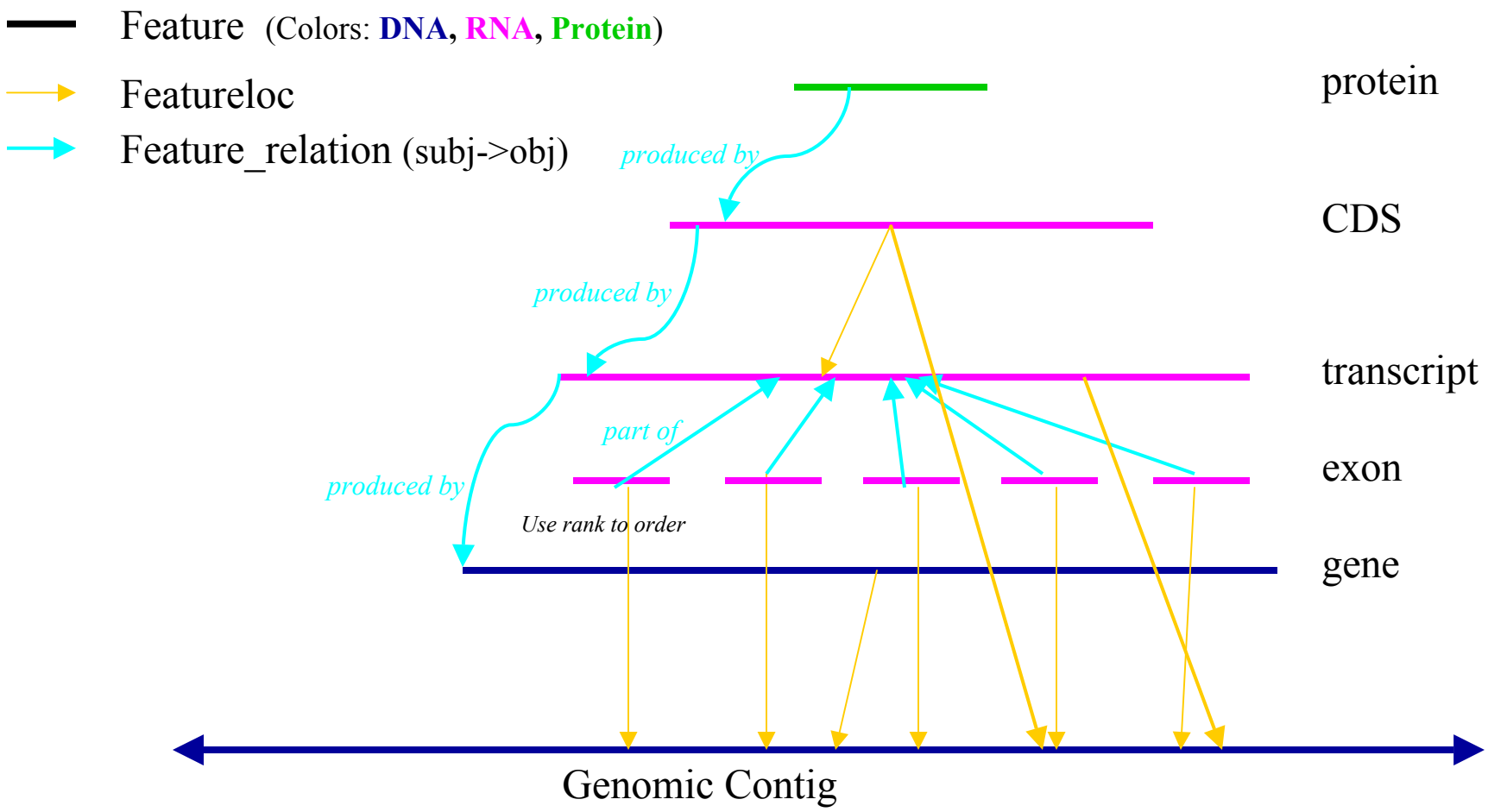
File Edit Plugins Filters Help

Sequence_Feature_Ontology

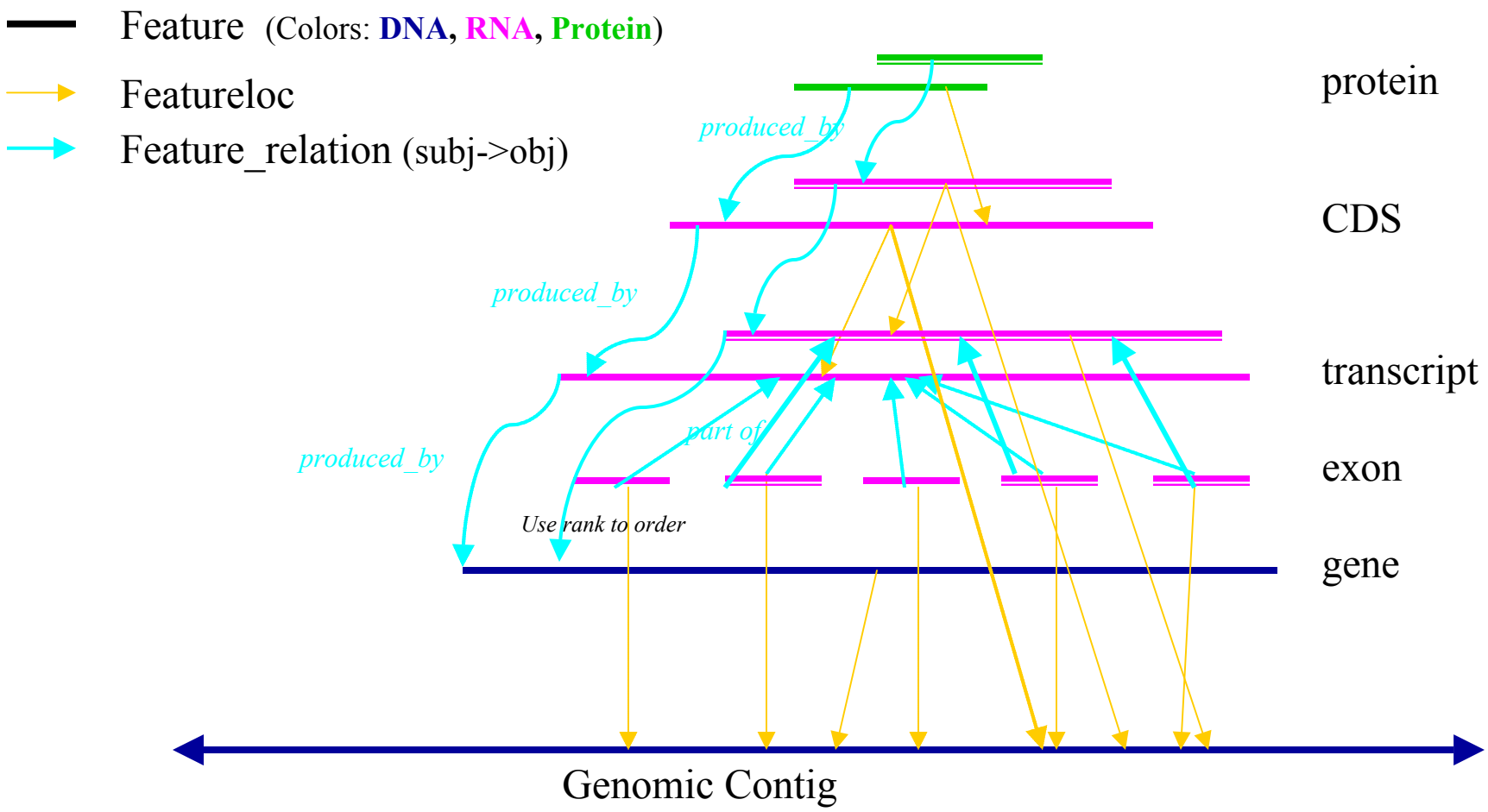
- chromosome variation
- consequences of mutation
- sequence feature
  - located sequence feature
    - annotation comment
    - gene element
      - regulatory region
      - transcript region
    - nucleotide motif
    - reagent
    - region
      - conserved region
      - deletion
      - exon
      - gene group
      - gene region
      - integrated virus
      - intron
      - match
      - non-gene feature
      - recombination feature
      - repeat region
      - replication feature
      - sequence secondary structure
      - sequence variation feature
    - site
  - sequence attribute

- region
  - conserved region
    - coding conserved region
    - noncoding conserved region
    - syntenic region
  - deletion
  - exon
    - 3'-exon
    - 5'-exon
    - coding exon
    - interior exon
    - noncoding exon
    - single exon
  - gene group
  - gene region
  - integrated virus
  - intron
    - 3'-intron
    - 5'-intron
    - autocatalytically spliced intron
    - branch site
    - group I intron
    - group II intron
    - interior intron
    - intronic splice enhancer
    - polypyrimidine tract
  - match
    - nucleotide to nucleotide match
    - nucleotide to protein match
    - protein to nucleotide match
    - protein to protein match

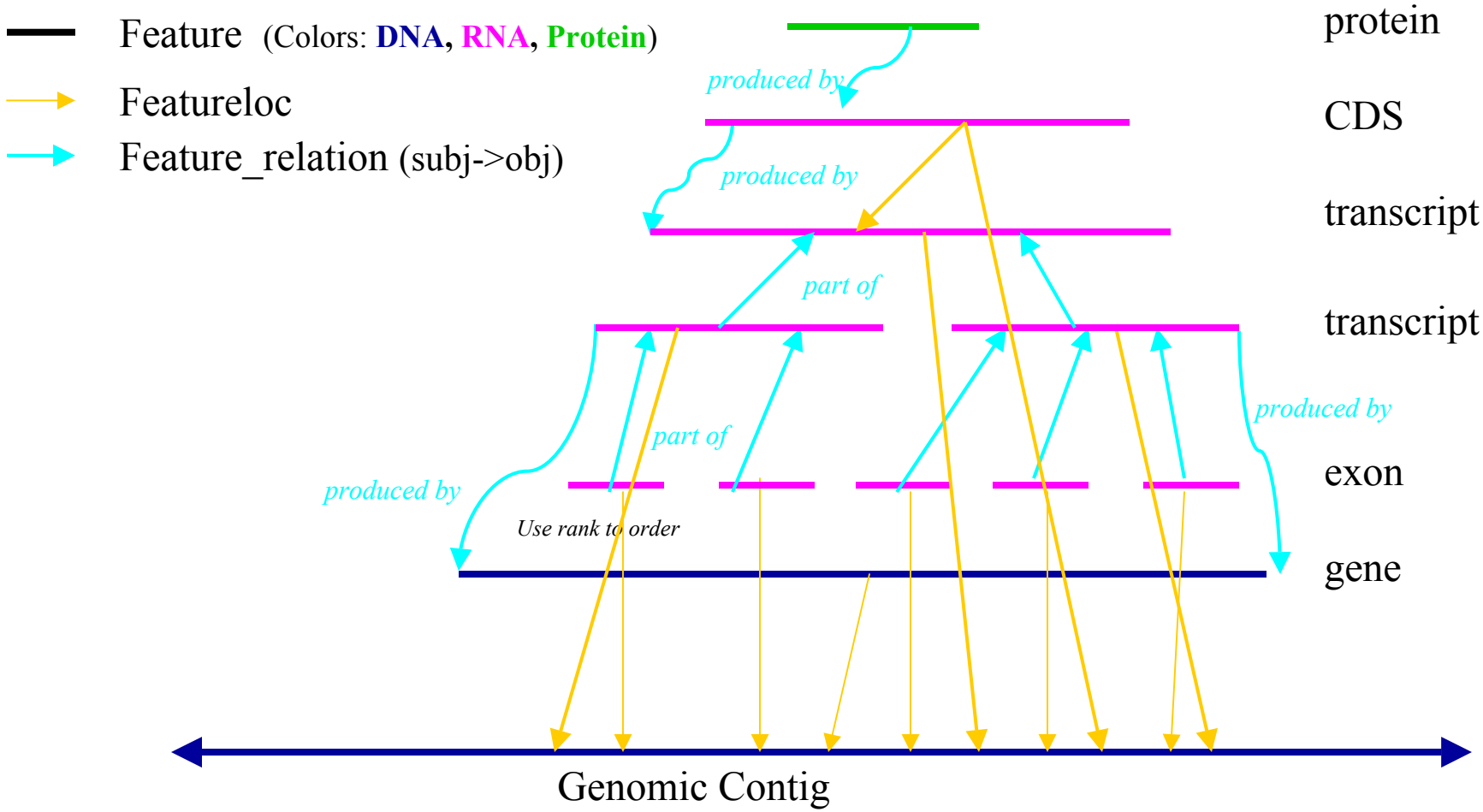
Central Dogma:
single spliced transcript



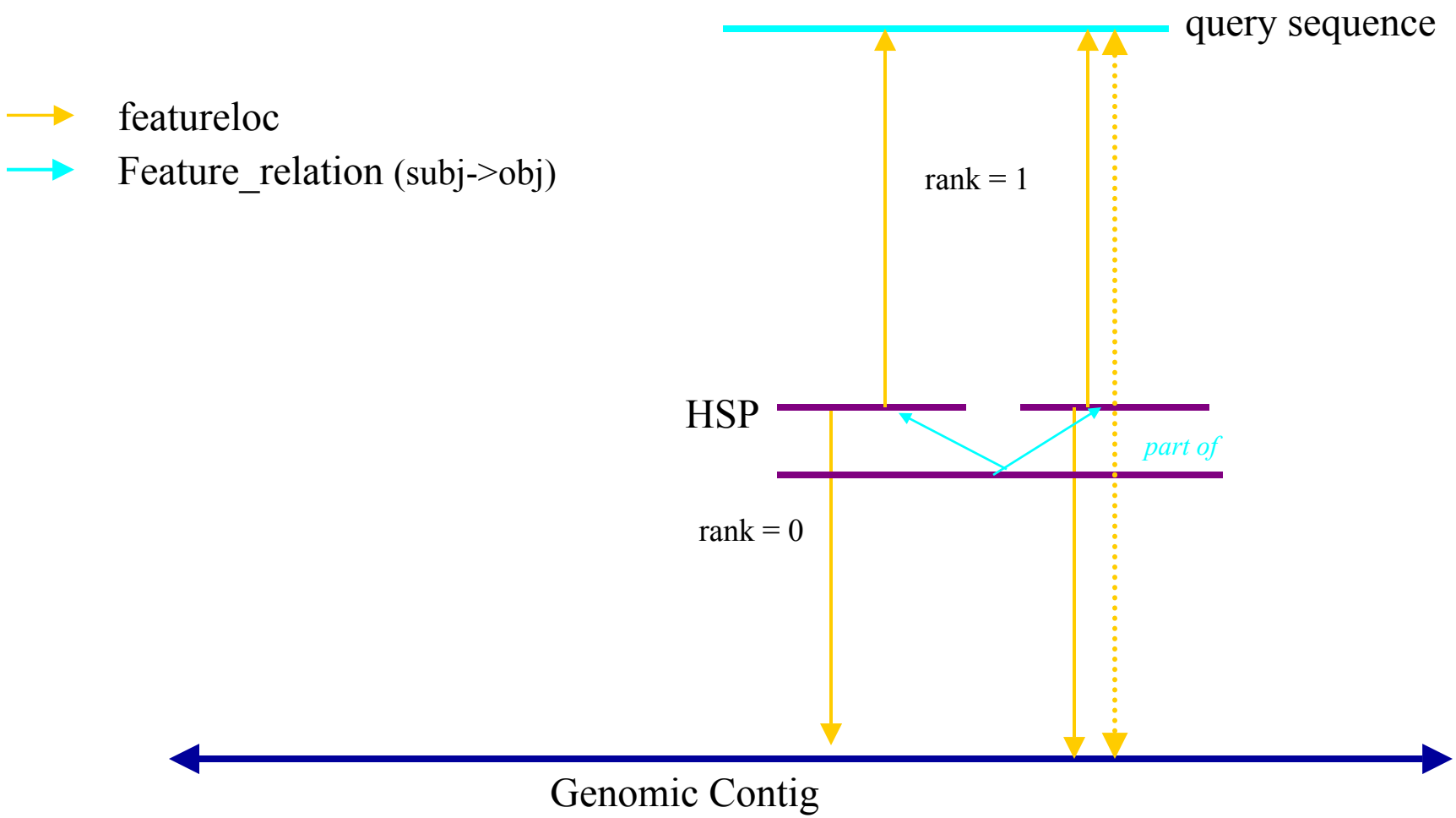
Central Dogma:
2nd transcript (alt. Splicing)



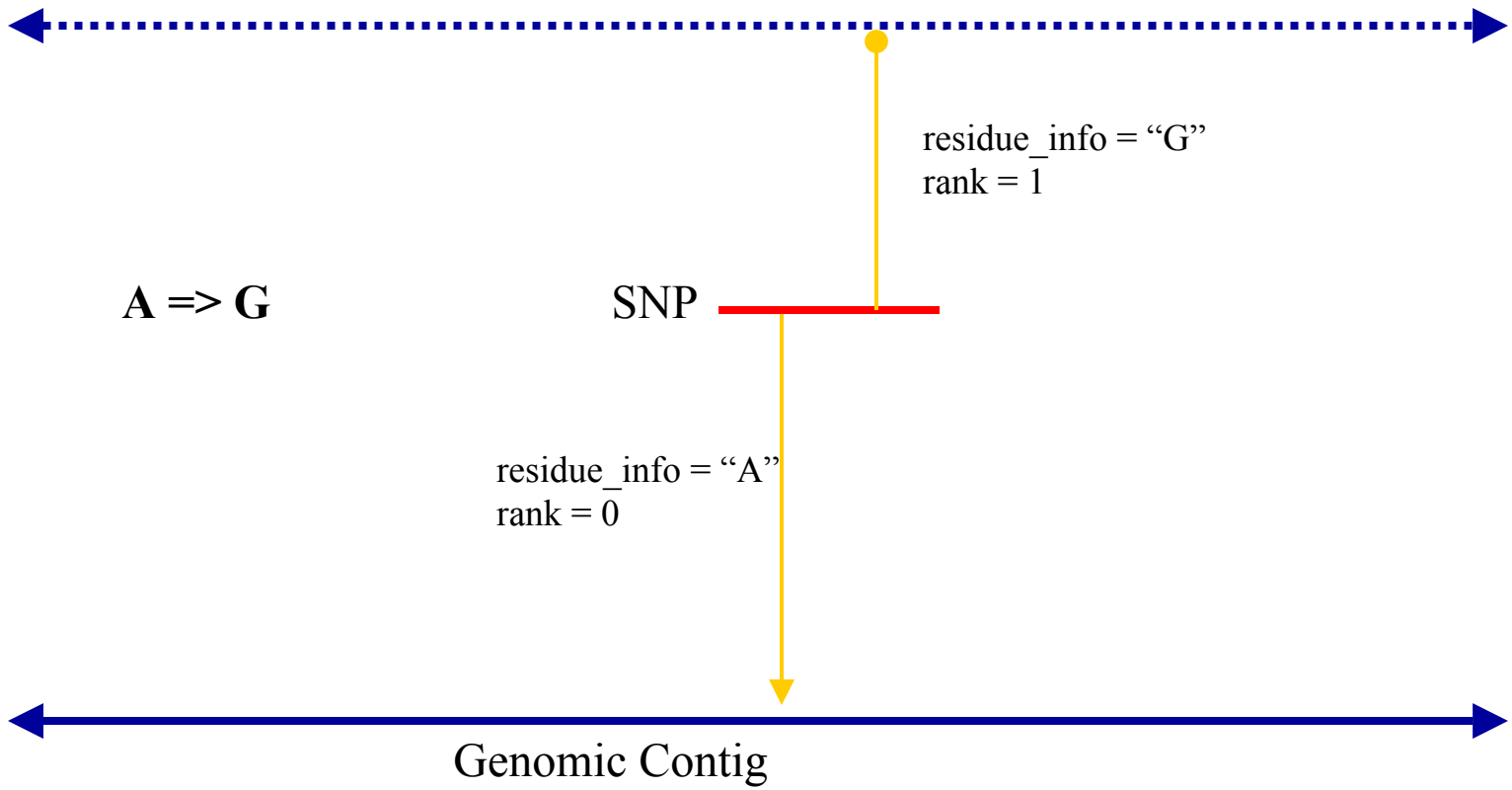
Pathological Cases:
trans-splicing



Pairwise Alignments

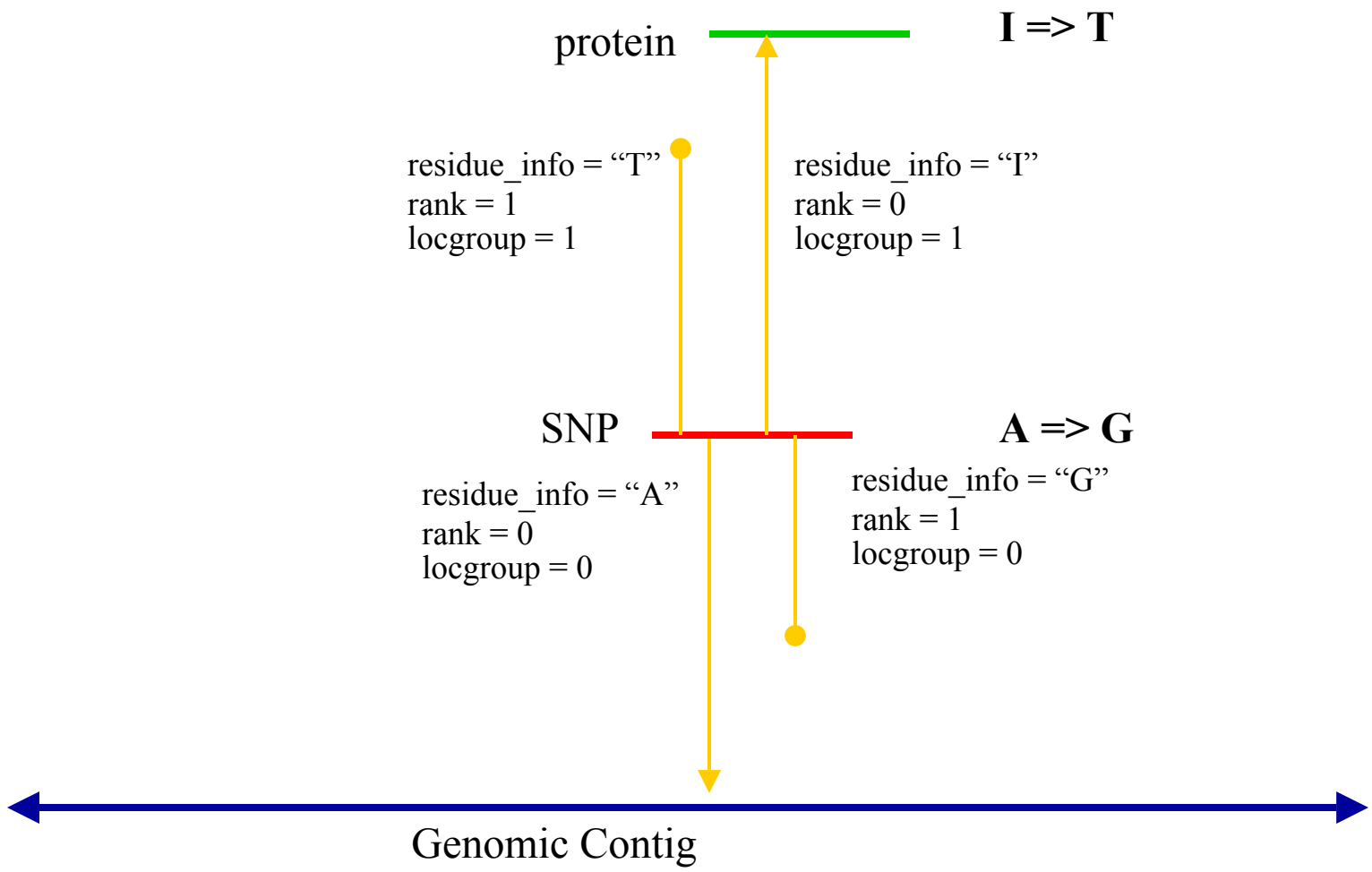


Sequence variations
SNPs



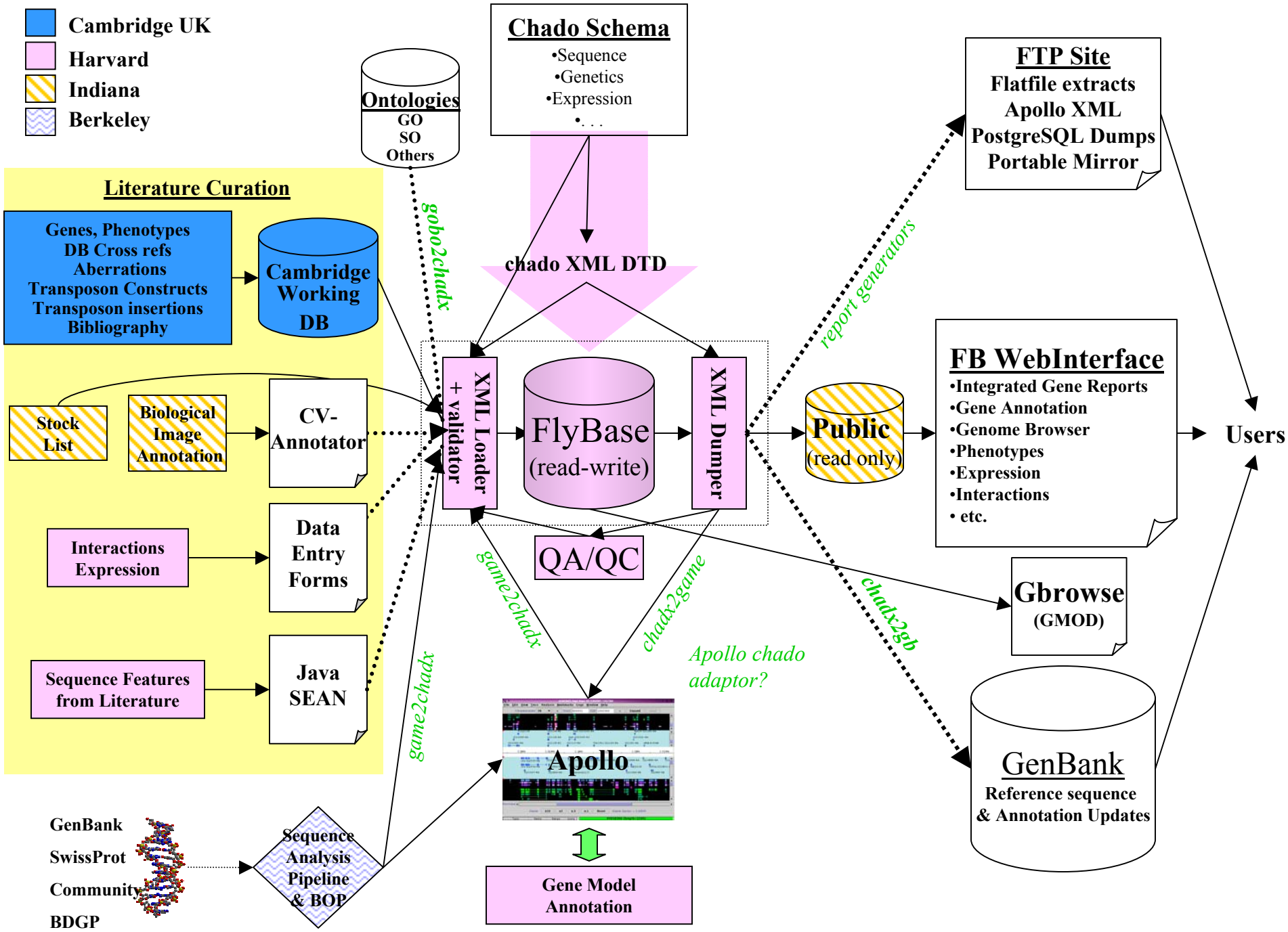
Sequence variations

SNPs (redundant mapping to protein)



Query Performance

- ROI Query
  - Reasonable, not stellar performance on PostGreSQL with index on (srcfeature_id, min, max)
  - Exploration of more sophisticated approaches yielded performance improvements in MySQL but not PostGreSQL
  - PostGreSQL functions simplify queries, e.g. select * from contains(*src,min,max*)
- Central Dogma Query
  - a few seconds for 3 levels, all info
  - 1 minute to include all overlapping features


Cambridge UK
Harvard
Indiana
Berkeley
Ontologies
GO
SO
Others
Chado Schema
•Sequence
•Genetics
•Expression
•. . .
FTP Site
Flatfile extracts
Apollo XML
PostgreSQL Dumps
Portable Mirror
Literature Curation
Genes, Phenotypes
DB Cross refs
Aberrations
Transposon Constructs
Transposon insertions
Bibliography
Cambridge Working DB
gobo2chadx
chado XML DTD
report generators
Stock List
Biological Image Annotation
CV-Annotator
XML Loader + validator
FlyBase
(read-write)
XML Dumper
Public
(read only)
FB WebInterface
•Integrated Gene Reports
•Gene Annotation
•Genome Browser
•Phenotypes
•Expression
•Interactions
• etc.
Users
Interactions Expression
Data Entry Forms
QA/QC
game2chadx
chadx2game
chadx2gb
Gbrowse
(GMOD)
Sequence Features from Literature
Java SEAN
game2chadx
Apollo chado adaptor?
Apollo
GenBank
Reference sequence
& Annotation Updates
GenBank
SwissProt
Community
BDGP
Sequence Analysis Pipeline & BOP
Gene Model Annotation

XORT

XML Object to Relational Translator

- Schema-driven tools
  - DTD generator: DDL -> DTD
    - also generates html, xml, .pl versions of schema
  - Validator:
    - **Not connected:**
      - **Syntax Verification: legal XML, correct element nesting**
      - **Some Semantic verification: NULLness, cardinality, local ID reference**
    - **Connected: reference validation**
    - **Loader-only: constraints, triggers**
  - Loader: XML -> DB
  - Dumper: DB -> XML
    - driven by XML "dumpspec"

Mapping XML to R-DBMS

- Policy#1: XML is independent of schema
  - Pro: ensures modularity, freedom to change one without the other (but why would you want to?)
  - Con: must maintain mapping when either changes / extends
- Policy#2: XML locked to schema
  - Pro: don't have to learn two things, mapping is frozen. Extension/revision is free.
  - Con: see Pro above.

XORT Mapping

- Elements
  - Table
  - Column (except primary key -- not visible in XML)
- Attributes
  - few and generic: transaction and reference control
- Element nesting
  - column within table
  - joined table within table -- joining column is implicit
  - foreign key table within foreign key column
- Modules
  - No module distinctions in chadoXML
- Limitations of DTD
  - Cardinality, NULLness, data type

```
<!-- ********************* TABLE feature ****************************
*    feature_id    serial    not null    primary key
*    dbxref_id    int    foreign key to dbxref.dbxref_id - dbxref_id here is intended for
the primary dbxref for this feature.  Additional dbxref links are made via feature_dbxref
*    organism_id    int    not null    foreign key to organism.organism_id    unique(2)
*    name    varchar(255) - the human-readable common name for a feature, for display
*    uniquename    text    not null    unique(2) - the unique name for a feature; may not
be particularly human-readable
*    residues    text
*    seqlen    int
*    md5checksum    char(32)
*    type_id    int    not null    foreign key to cvterm.cvterm_id
*    timeaccessioned    timestamp    not null    default current_timestamp -
timeaccessioned and timelastmodified are for handling object accession/  modification
timestamps (as opposed to db auditing info, handled elsewhere).  The expectation is that
these fields would be available to software  interacting with chado.
*    timelastmodified    timestamp    not null    default current_timestamp
     ************************************************************************** -->
<!ELEMENT feature (dbxref_id | organism_id | name | uniquename | residues | seqlen |
md5checksum | type_id | timeaccessioned | timelastmodified)*
    ( feature_cvterm | analysis | featureloc | featureprop | analysisfeature |
feature_genotype | featurepos | feature_dbxref | feature_synonym | featurerange |
feature_relationship | feature_phenotype | feature_pub | interaction_subj |
interaction_obj | feature_expression)*
)>

<!ATTLIST feature
    id CDATA #IMPLIED
    ref CDATA #IMPLIED
    op (lookup | insert | update | force | delete) #IMPLIED>
<!ELEMENT dbxref_id ( #PCDATA | dbxref ) >    <!ATTLIST dbxref_id op (update) #IMPLIED>
<!ELEMENT organism_id ( #PCDATA | organism ) >    <!ATTLIST organism_id op (update)
#IMPLIED>
<!ELEMENT name #PCDATA >    <!ATTLIST name op (update) #IMPLIED>
<!ELEMENT uniquename #PCDATA >    <!ATTLIST uniquename op (update) #IMPLIED>
<!ELEMENT residues #PCDATA >    <!ATTLIST residues op (update) #IMPLIED>
<!ELEMENT seqlen #PCDATA >    <!ATTLIST seqlen op (update) #IMPLIED>
<!ELEMENT md5checksum #PCDATA >    <!ATTLIST md5checksum op (update) #IMPLIED>
<!ELEMENT type_id ( #PCDATA | cvterm ) >    <!ATTLIST type_id op (update) #IMPLIED>
<!ELEMENT timeaccessioned #PCDATA >    <!ATTLIST timeaccessioned op (update) #IMPLIED>
<!ELEMENT timelastmodified #PCDATA >    <!ATTLIST timelastmodified op (update) #IMPLIED>
```

Object References

How to refer to persistent objects within XML?
(a.k.a. foreign key columns)

- **By Unique Key Value(s)**
  - object can be in XML file or DB
- **By local ID**
  - only for references to objects in same XML file
  - need not be in DB
  - local ID can be any symbol - def before ref
  - reduces duplication within XML
- **By Global accession**
  - currently only for feature
  - simple extension mechanism using Perl fragments

Object Reference

by key values

```
<foreign_key_col>
        <primarytable>
           <keycol1>keyval1</keycol1>
           ... more key cols if needed
        </primarytable>
</foreign_key_col>
```

E.g.

```
    <feature>
        <type_id>
            <cvterm>
                <cv_id>
                    <cv>
                        <name>Sequence Ontology</name>
                    </cv>
                </cv_id>
                <name>exon</name>
            </cvterm>
        </type_id>
      ....
```

Object Reference
by Local ID

```
<cv id="SO">
    <name>Sequence Ontology</name>
</cv>

<cvterm id="exon">
     <cv_id>SO</cv_id>
    <name>exon</name>
</cvterm>

<feature>
    <type_id>exon</type_id>
  ...
```

Object Reference
by Global Accession

```
<feature_relationship>
  <subjfeature_id>GB:g012345
      </subjfeature_id>
  ...
```

Transactions

- **Lookup:** <table op="lookup">...
- **Insert:** <table op="insert">...
- **Delete:**

```
<table op="delete">
    <keycol1>val1</keycol1>…
```

- **Update**

```
<table op="update">
    <keycol1>val1</keycol1>
    <keycol2>val2</keycol1>
    <keycol1>newval</keycol1>
```

- **Force:** <table op="force">...

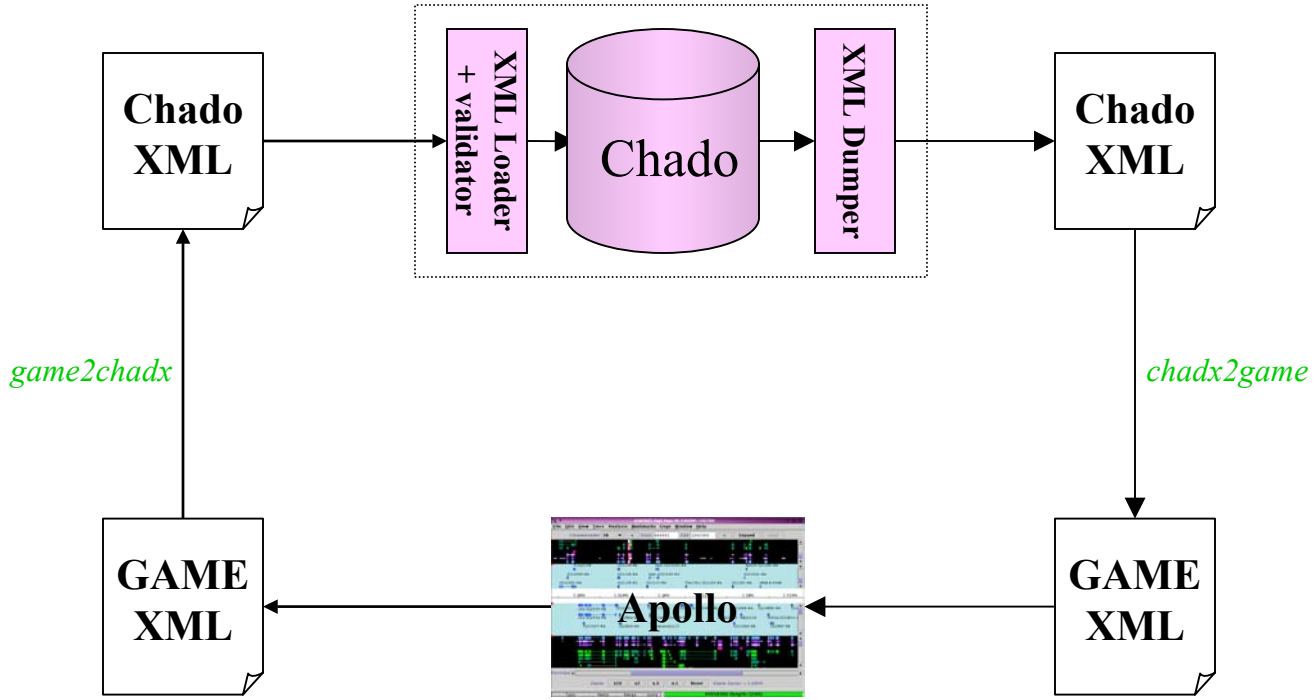
  Combination of lookup, insert and update

Dumper
XML-driven extraction

- Default behavior: given an object class and ID, dump all direct values and linktables, with refs to foreign keys.
- Nondefault behavior specified by XML dumpspecs using same DTD with a few additions:
  - attribute dump= all | cols | select | none
  - attribute test = yes | no
  - element OR
  - element _sql
  - element _appdata
- Workaround with views, _sql
- Current use cases:
  - Dump a gene for a gene detail page
  - Dump a scaffold for Apollo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE chado SYSTEM "/users/zhou/work/flybase/xml/chado_stan.dtd">
<!-- 1. dump all information for gene CG9570 and all information for transcript, all for
translation, for feature_evidence, dump all cols of foreign object:featureloc -->
<chado>
…
<feature>
        <uniquename test="yes"><or>CG3665</or>
                        <or>CG3139</or><or>CG3497</or></uniquename>
        <!-- get all mRNA of those genes -->
        <feature_relationship dump="all">
            <subjfeature_id test="yes">
                <feature>
                    <type_id>mRNA</type_id>
                </feature>
        </subjfeature_id>
        <subjfeature_id>
                <feature>
                <!-- get all exons of those mRNA -->
                    <feature_relationship>
                        <subjfeature_id test="yes">
                            <feature>
                                <type_id>exon</type_id>
                ….
```

Chado <-> Apollo Interaction



DUMPER Concerns

- Expressivity
- Speed
- XML file size
- Memory

What's next

- Debug Apollo / chado roundtrip
- CV issues
  - Hierarchical queries
  - SO compliance
  - feature relationship types
- Schema extensions
  - genetics module - review in Fall
  - expression
  - Function generally
- UI development - not based on dumper

Architectural Principles

- Semi-permeable XML layer:
  - support but do not require XML
- Fix mapping, let schema vary
- Plan for schema evolution -- schema-driven tools
- Course-grained coupling of modules made possible by XML standardization

GMOD Architecture:
oxymoron?

- Claim: self-contained, interoperable reusable complete system will not happen by itself
- All tools speak all formats?
- Standardize DB, API, or format?
  - Format standardization good for persistence, less good for querying
  - Bless existing format: ASN.1, GFF, BSML, SBML, … or YAF?

Credits

Pinglei Zhou - loader, dumper, XML design
Frank Smutniak - game2chadx, chadx2game
Colin Wiel - gadfly2chado migration, schema
David Emmert - schema, migration
Chris Mungall, Suzi Lewis - schema, SO
Stan Letovsky - XML/tool design, dtd generator
Susan Russo, Mark Zytkovicz - PostGreSQL
Don Gilbert - XML customer
Scott Cain - GBROWSE/Chado
Allen Day - schema (expression)
Hilmar Lapp - ROI query optimization
...