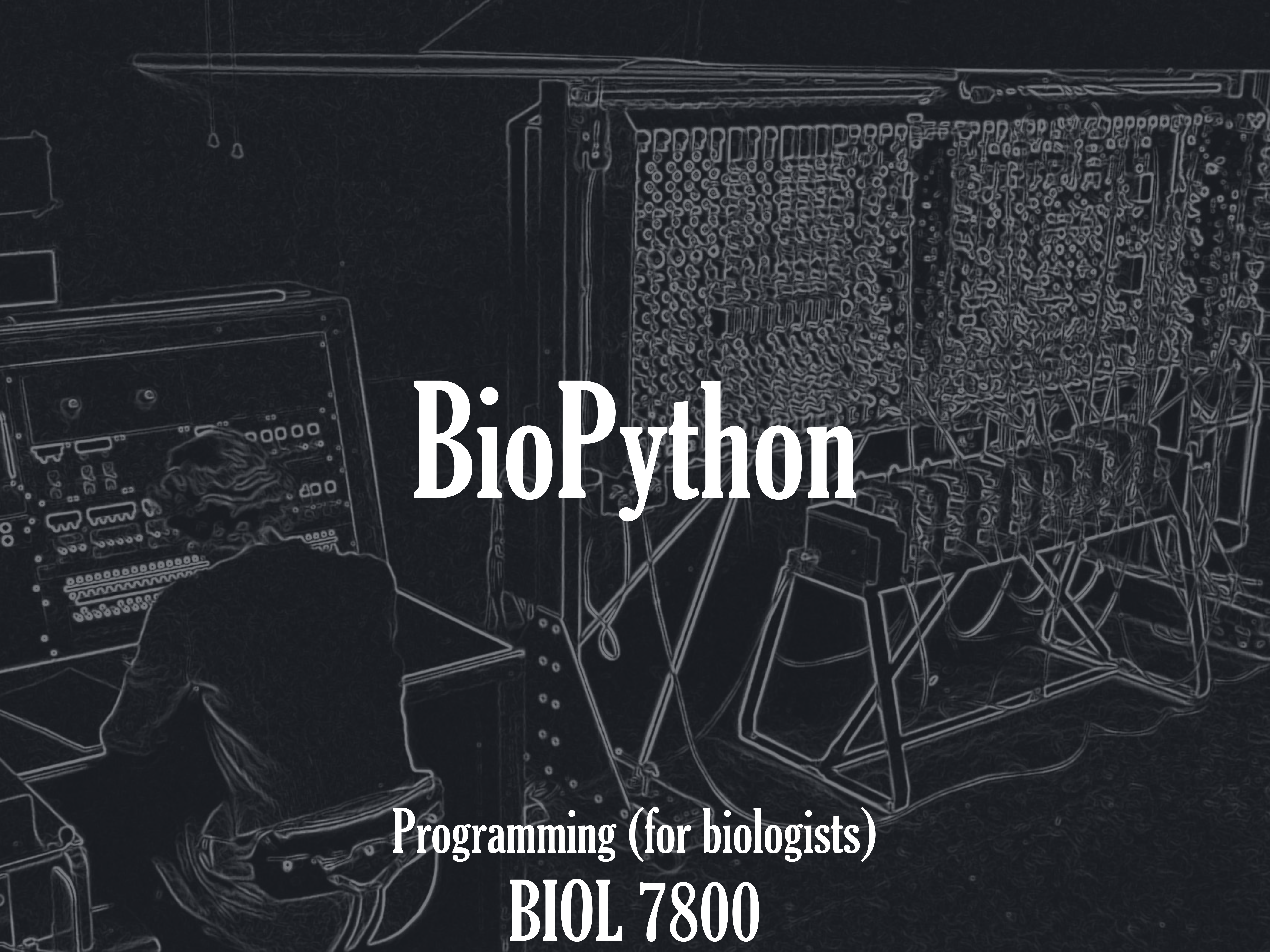


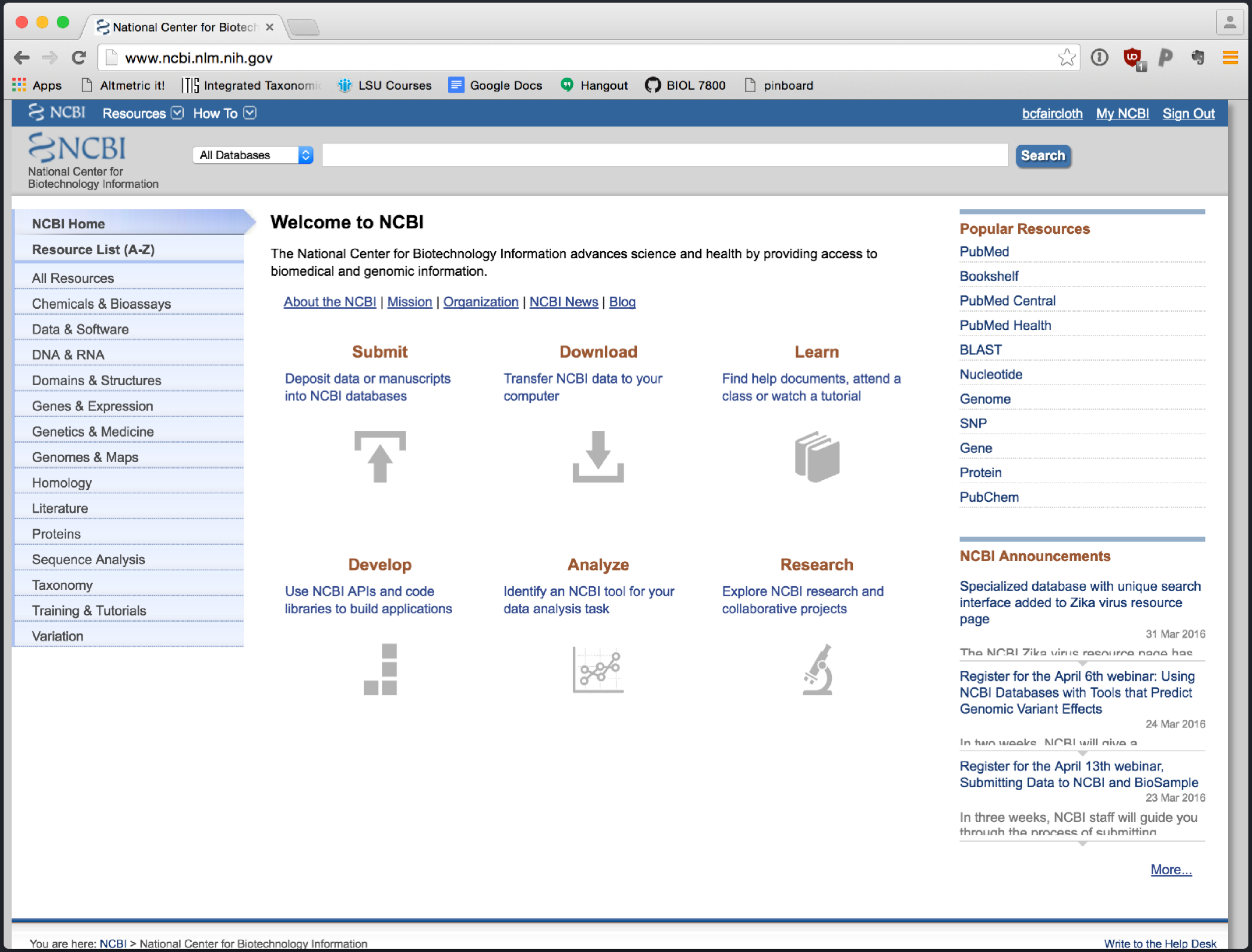
BioPython

Programming (for biologists)

BIOL 7800

NCBI

National Center for Biotechnology Information



- NCBI Home
- Resource List (A-Z)
- All Resources
- Chemicals & Bioassays
- Data & Software
- DNA & RNA
- Domains & Structures
- Genes & Expression
- Genetics & Medicine
- Genomes & Maps
- Homology
- Literature
- Proteins
- Sequence Analysis
- Taxonomy
- Training & Tutorials
- Variation

Welcome to NCBI

The National Center for Biotechnology Information advances science and health by providing access to biomedical and genomic information.

About the NCBI | Mission | Organization | NCBI News | Blog

Submit

Deposit data or manuscripts into NCBI databases

Download

Transfer NCBI data to your computer

Learn

Find help documents, attend a class or watch a tutorial

Develop

Use NCBI APIs and code libraries to build applications

Analyze

Identify an NCBI tool for your data analysis task

Research

Explore NCBI research and collaborative projects

Popular Resources

- PubMed
- Bookshelf
- PubMed Central
- PubMed Health
- BLAST
- Nucleotide
- Genome
- SNP
- Gene
- Protein
- PubChem

NCBI Announcements

Specialized database with unique search interface added to Zika virus resource page
31 Mar 2016
The NCBI Zika virus resource page has

Register for the April 6th webinar: Using NCBI Databases with Tools that Predict Genomic Variant Effects
24 Mar 2016
In two weeks, NCBI will give a

Register for the April 13th webinar, Submitting Data to NCBI and BioSample
23 Mar 2016
In three weeks, NCBI staff will guide you through the process of submitting

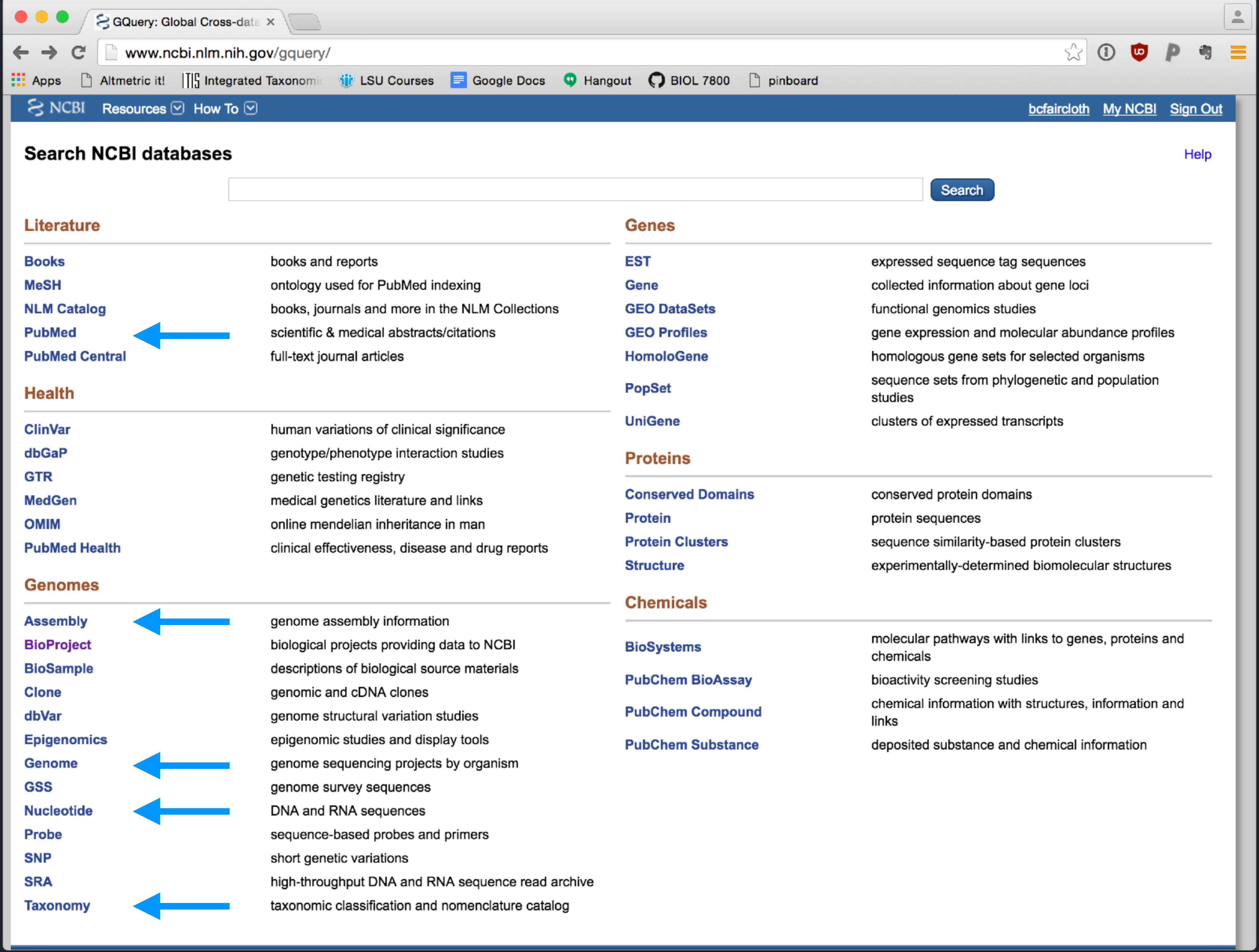
More...

You are here: NCBI > National Center for Biotechnology Information

Write to the Help Desk

NCBI

National Center for Biotechnology Information



Search NCBI databases

Literature

Books	books and reports
MeSH	ontology used for PubMed indexing
NLM Catalog	books, journals and more in the NLM Collections
PubMed ←	scientific & medical abstracts/citations
PubMed Central	full-text journal articles

Health

ClinVar	human variations of clinical significance
dbGaP	genotype/phenotype interaction studies
GTR	genetic testing registry
MedGen	medical genetics literature and links
OMIM	online mendelian inheritance in man
PubMed Health	clinical effectiveness, disease and drug reports

Genomes

Assembly ←	genome assembly information
BioProject	biological projects providing data to NCBI
BioSample	descriptions of biological source materials
Clone	genomic and cDNA clones
dbVar	genome structural variation studies
Epigenomics	epigenomic studies and display tools
Genome ←	genome sequencing projects by organism
GSS	genome survey sequences
Nucleotide ←	DNA and RNA sequences
Probe	sequence-based probes and primers
SNP	short genetic variations
SRA	high-throughput DNA and RNA sequence read archive
Taxonomy ←	taxonomic classification and nomenclature catalog

Genes

EST	expressed sequence tag sequences
Gene	collected information about gene loci
GEO DataSets	functional genomics studies
GEO Profiles	gene expression and molecular abundance profiles
HomoloGene	homologous gene sets for selected organisms
PopSet	sequence sets from phylogenetic and population studies
UniGene	clusters of expressed transcripts

Proteins

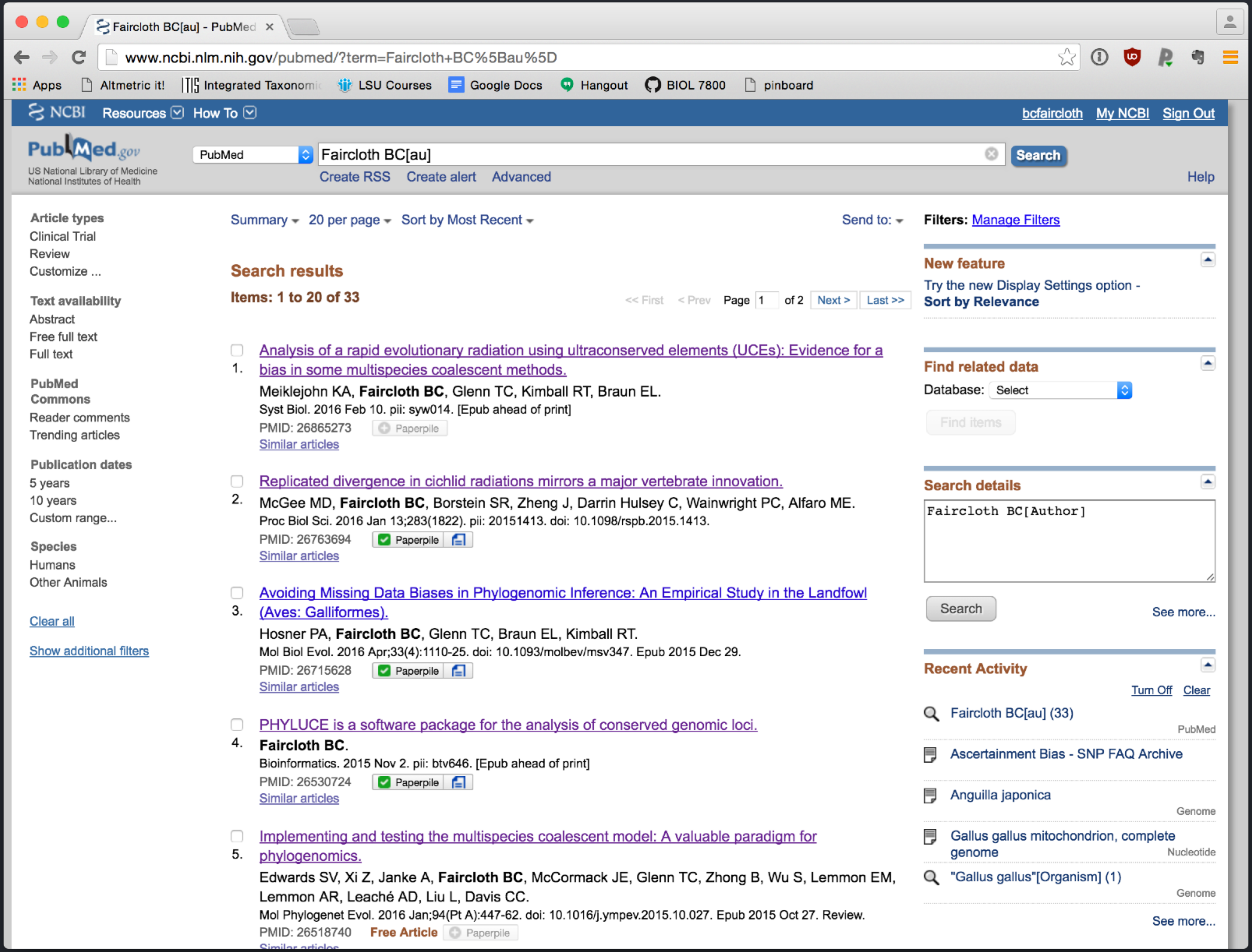
Conserved Domains	conserved protein domains
Protein	protein sequences
Protein Clusters	sequence similarity-based protein clusters
Structure	experimentally-determined biomolecular structures

Chemicals

BioSystems	molecular pathways with links to genes, proteins and chemicals
PubChem BioAssay	bioactivity screening studies
PubChem Compound	chemical information with structures, information and links
PubChem Substance	deposited substance and chemical information

Many different databases

NCBI | PubMed



Faircloth BC[au]

Search results

Items: 1 to 20 of 33

☐ 1. Analysis of a rapid evolutionary radiation using ultraconserved elements (UCEs): Evidence for a bias in some multispecies coalescent methods.
Meiklejohn KA, **Faircloth BC**, Glenn TC, Kimball RT, Braun EL.
Syst Biol. 2016 Feb 10. pii: syw014. [Epub ahead of print]
PMID: 26865273 Paperpile
Similar articles

☐ 2. Replicated divergence in cichlid radiations mirrors a major vertebrate innovation.
McGee MD, **Faircloth BC**, Borstein SR, Zheng J, Darrin Hulsey C, Wainwright PC, Alfaro ME.
Proc Biol Sci. 2016 Jan 13;283(1822). pii: 20151413. doi: 10.1098/rspb.2015.1413.
PMID: 26763694 ✓ Paperpile
Similar articles

☐ 3. Avoiding Missing Data Biases in Phylogenomic Inference: An Empirical Study in the Landfowl (Aves: Galliformes).
Hosner PA, **Faircloth BC**, Glenn TC, Braun EL, Kimball RT.
Mol Biol Evol. 2016 Apr;33(4):1110-25. doi: 10.1093/molbev/msv347. Epub 2015 Dec 29.
PMID: 26715628 ✓ Paperpile
Similar articles

☐ 4. PHYLUCE is a software package for the analysis of conserved genomic loci.
Faircloth BC.
Bioinformatics. 2015 Nov 2. pii: btv646. [Epub ahead of print]
PMID: 26530724 ✓ Paperpile
Similar articles

☐ 5. Implementing and testing the multispecies coalescent model: A valuable paradigm for phylogenomics.
Edwards SV, Xi Z, Janke A, **Faircloth BC**, McCormack JE, Glenn TC, Zhong B, Wu S, Lemmon EM, Lemmon AR, Leaché AD, Liu L, Davis CC.
Mol Phylogenet Evol. 2016 Jan;94(Pt A):447-62. doi: 10.1016/j.ympev.2015.10.027. Epub 2015 Oct 27. Review.
PMID: 26518740 Free Article Paperpile

New feature

Try the new Display Settings option - **Sort by Relevance**

Search details

```
Faircloth BC[Author]
```

Faircloth BC[au] (33) PubMed

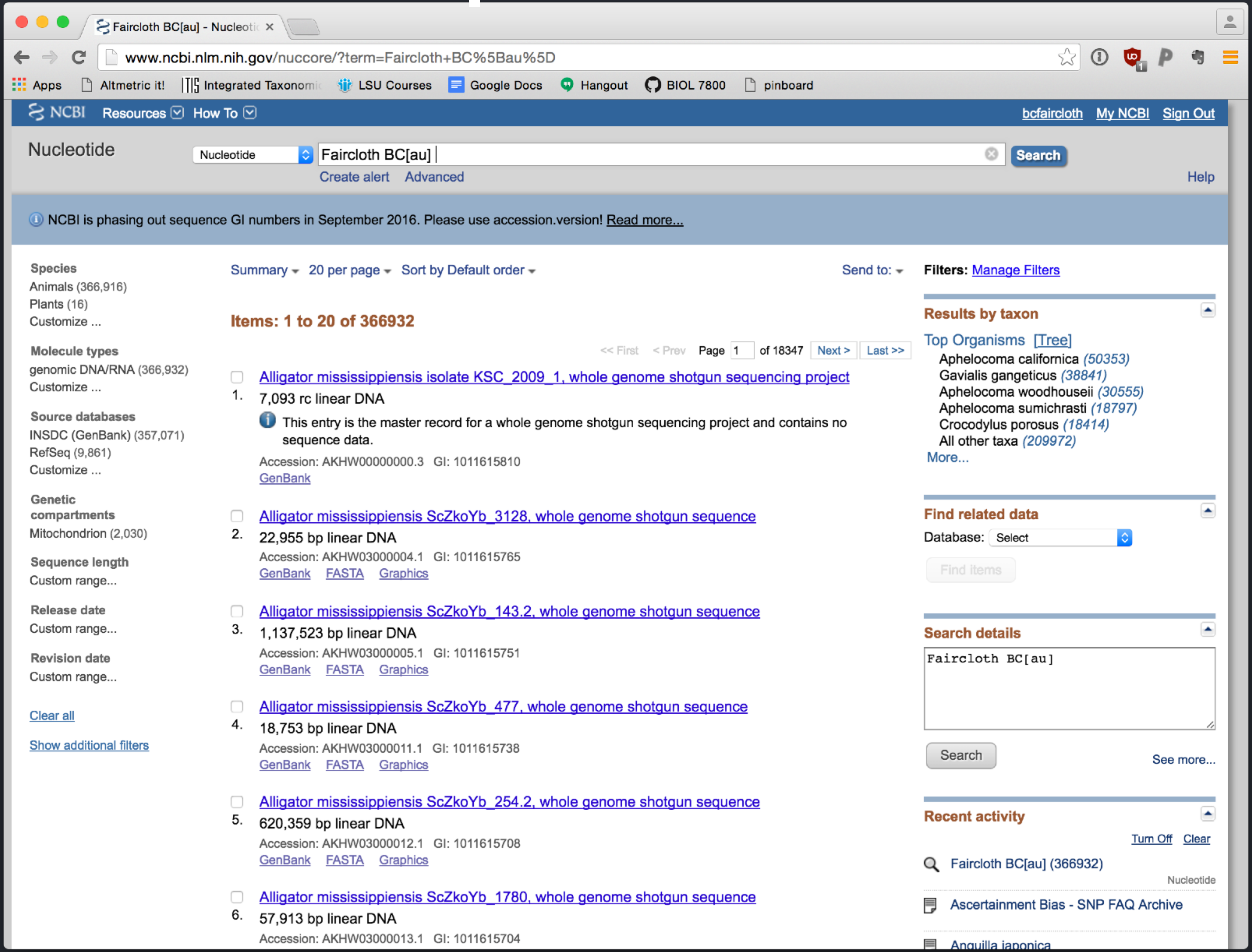
Ascertainment Bias - SNP FAQ Archive

Anguilla japonica Genome

Gallus gallus mitochondrion, complete genome Nucleotide

"Gallus gallus"[Organism] (1) Genome

NCBI | Nucleotide



Nucleotide ⌄ Faircloth BC[au]

NCBI is phasing out sequence GI numbers in September 2016. Please use accession.version! Read more...

Species
Animals (366,916)
Plants (16)
Customize ...

Molecule types
genomic DNA/RNA (366,932)
Customize ...

Source databases
INSDC (GenBank) (357,071)
RefSeq (9,861)
Customize ...

Genetic compartments
Mitochondrion (2,030)

Sequence length
Custom range...

Release date
Custom range...

Revision date
Custom range...

Clear all

Show additional filters

Summary ▾ 20 per page ▾ Sort by Default order ▾ Send to: ▾

Items: 1 to 20 of 366932

<< First < Prev Page 1 of 18347 Next > Last >>

☐ 1. Alligator mississippiensis isolate KSC_2009_1, whole genome shotgun sequencing project
7,093 rc linear DNA
This entry is the master record for a whole genome shotgun sequencing project and contains no sequence data.
Accession: AKHW00000000.3 GI: 1011615810
GenBank

☐ 2. Alligator mississippiensis ScZkoYb_3128, whole genome shotgun sequence
22,955 bp linear DNA
Accession: AKHW03000004.1 GI: 1011615765
GenBank FASTA Graphics

☐ 3. Alligator mississippiensis ScZkoYb_143.2, whole genome shotgun sequence
1,137,523 bp linear DNA
Accession: AKHW03000005.1 GI: 1011615751
GenBank FASTA Graphics

☐ 4. Alligator mississippiensis ScZkoYb_477, whole genome shotgun sequence
18,753 bp linear DNA
Accession: AKHW03000011.1 GI: 1011615738
GenBank FASTA Graphics

☐ 5. Alligator mississippiensis ScZkoYb_254.2, whole genome shotgun sequence
620,359 bp linear DNA
Accession: AKHW03000012.1 GI: 1011615708
GenBank FASTA Graphics

☐ 6. Alligator mississippiensis ScZkoYb_1780, whole genome shotgun sequence
57,913 bp linear DNA
Accession: AKHW03000013.1 GI: 1011615704

Filters: Manage Filters

Results by taxon

Top Organisms [Tree]

Aphelocoma californica	*(50353)*
Gavialis gangeticus	*(38841)*
Aphelocoma woodhouseii	*(30555)*
Aphelocoma sumichrasti	*(18797)*
Crocodylus porosus	*(18414)*
All other taxa	*(209972)*

More...

Find related data

Database: Select

Find items

Search details

```
Faircloth BC[au]
```

Search

See more...

Recent activity

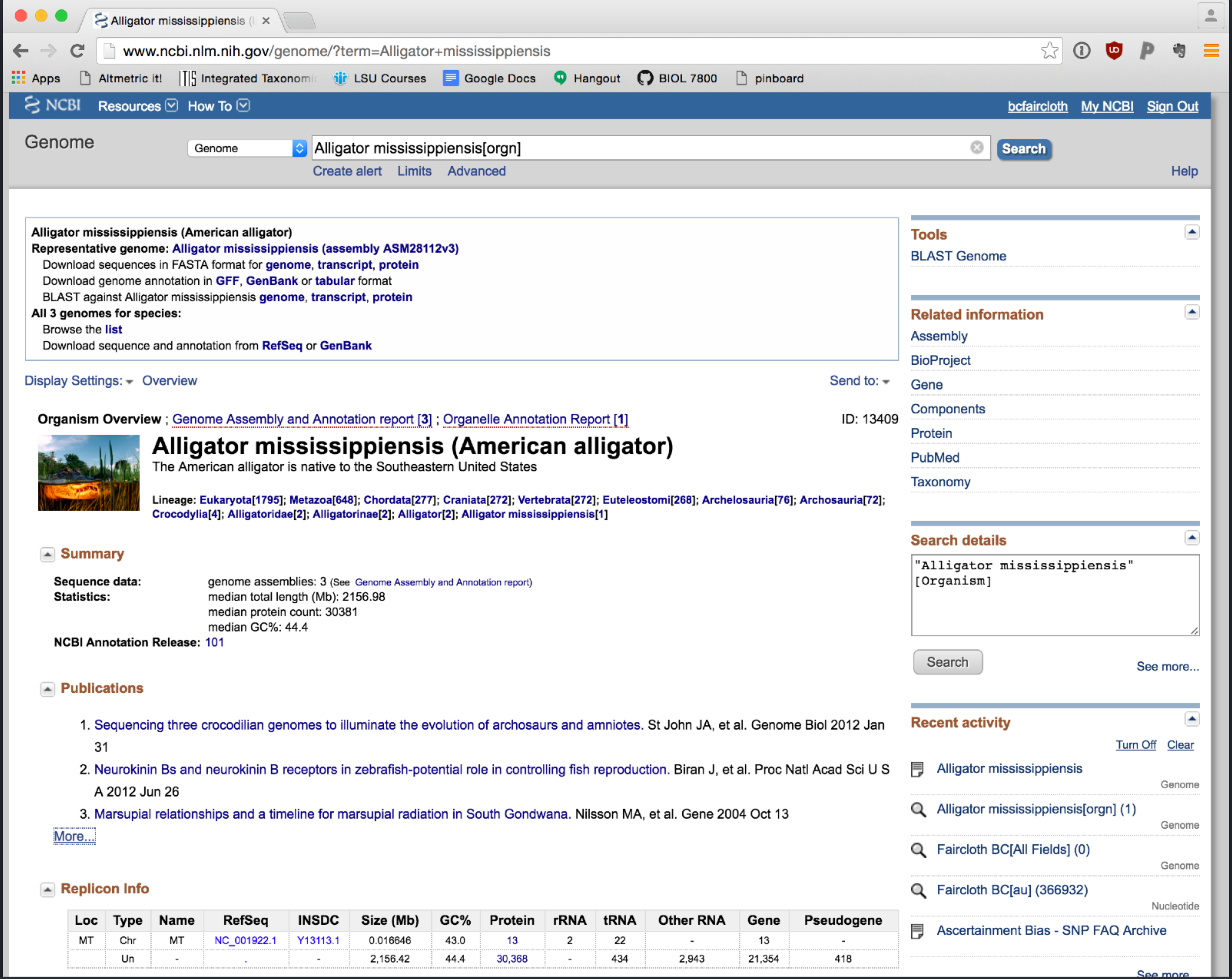
Turn Off Clear

Faircloth BC[au] (366932)
Nucleotide

Ascertainment Bias - SNP FAQ Archive

Anguilla japonica

NCBI | Genome



Alligator mississippiensis (American alligator)
Representative genome: Alligator mississippiensis (assembly ASM28112v3)
Download sequences in FASTA format for **genome**, **transcript**, **protein**
Download genome annotation in **GFF**, **GenBank** or **tabular** format
BLAST against Alligator mississippiensis **genome**, **transcript**, **protein**
All 3 genomes for species:
Browse the **list**
Download sequence and annotation from **RefSeq** or **GenBank**

Display Settings: Overview — Send to:

Organism Overview ; Genome Assembly and Annotation report [3] ; Organelle Annotation Report [1] — ID: 13409

Alligator mississippiensis (American alligator)

The American alligator is native to the Southeastern United States

Lineage: Eukaryota[1795]; Metazoa[648]; Chordata[277]; Craniata[272]; Vertebrata[272]; Euteleostomi[268]; Archelosauria[76]; Archosauria[72]; Crocodylia[4]; Alligatoridae[2]; Alligatorinae[2]; Alligator[2]; Alligator mississippiensis[1]

Summary

Sequence data: genome assemblies: 3 (See Genome Assembly and Annotation report)
Statistics: median total length (Mb): 2156.98
median protein count: 30381
median GC%: 44.4
NCBI Annotation Release: 101

Publications

1. Sequencing three crocodilian genomes to illuminate the evolution of archosaurs and amniotes. St John JA, et al. Genome Biol 2012 Jan 31
2. Neurokinin Bs and neurokinin B receptors in zebrafish-potential role in controlling fish reproduction. Biran J, et al. Proc Natl Acad Sci U S A 2012 Jun 26
3. Marsupial relationships and a timeline for marsupial radiation in South Gondwana. Nilsson MA, et al. Gene 2004 Oct 13

More...

Replicon Info

Loc	Type	Name	RefSeq	INSDC	Size (Mb)	GC%	Protein	rRNA	tRNA	Other RNA	Gene	Pseudogene
MT	Chr	MT	NC_001922.1	Y13113.1	0.016646	43.0	13	2	22	-	13	-
	Un	-	.	-	2,156.42	44.4	30,368	-	434	2,943	21,354	418

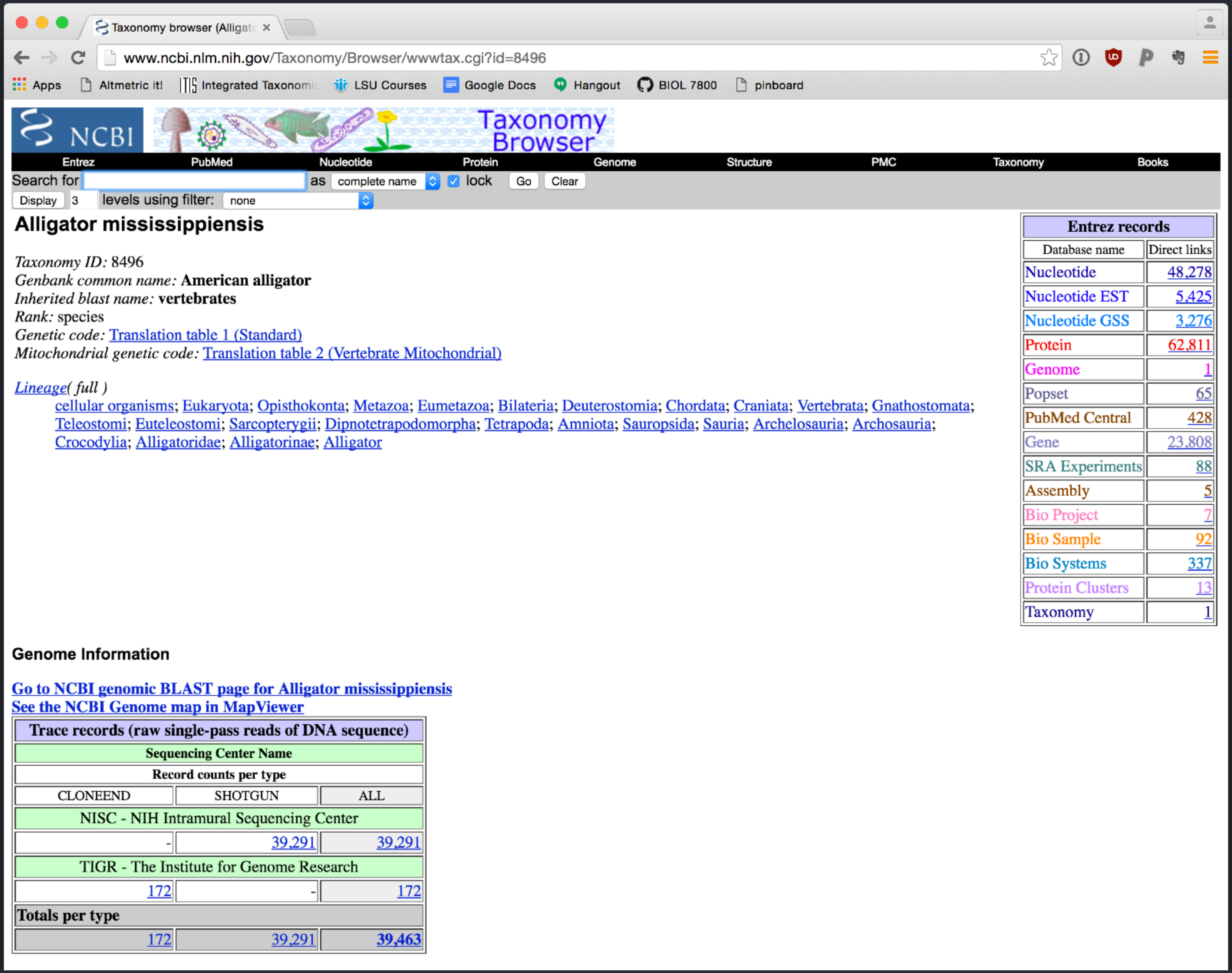
Tools
BLAST Genome

Related information
Assembly
BioProject
Gene
Components
Protein
PubMed
Taxonomy

Search details
```
"Alligator mississippiensis"
[Organism]
```
Search — See more...

Recent activity
Turn Off Clear
Alligator mississippiensis — Genome
Alligator mississippiensis[orgn] (1) — Genome
Faircloth BC[All Fields] (0) — Genome
Faircloth BC[au] (366932) — Nucleotide
Ascertainment Bias - SNP FAQ Archive

NCBI | Taxonomy



Alligator mississippiensis

Taxonomy ID: 8496
Genbank common name: **American alligator**
Inherited blast name: **vertebrates**
Rank: species
Genetic code: Translation table 1 (Standard)
Mitochondrial genetic code: Translation table 2 (Vertebrate Mitochondrial)

Lineage(*full*)
cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Deuterostomia; Chordata; Craniata; Vertebrata; Gnathostomata; Teleostomi; Euteleostomi; Sarcopterygii; Dipnotetrapodomorpha; Tetrapoda; Amniota; Sauropsida; Sauria; Archelosauria; Archosauria; Crocodylia; Alligatoridae; Alligatorinae; Alligator

Entrez records	
Database name	Direct links
Nucleotide	48,278
Nucleotide EST	5,425
Nucleotide GSS	3,276
Protein	62,811
Genome	1
Popset	65
PubMed Central	428
Gene	23,808
SRA Experiments	88
Assembly	5
Bio Project	7
Bio Sample	92
Bio Systems	337
Protein Clusters	13
Taxonomy	1

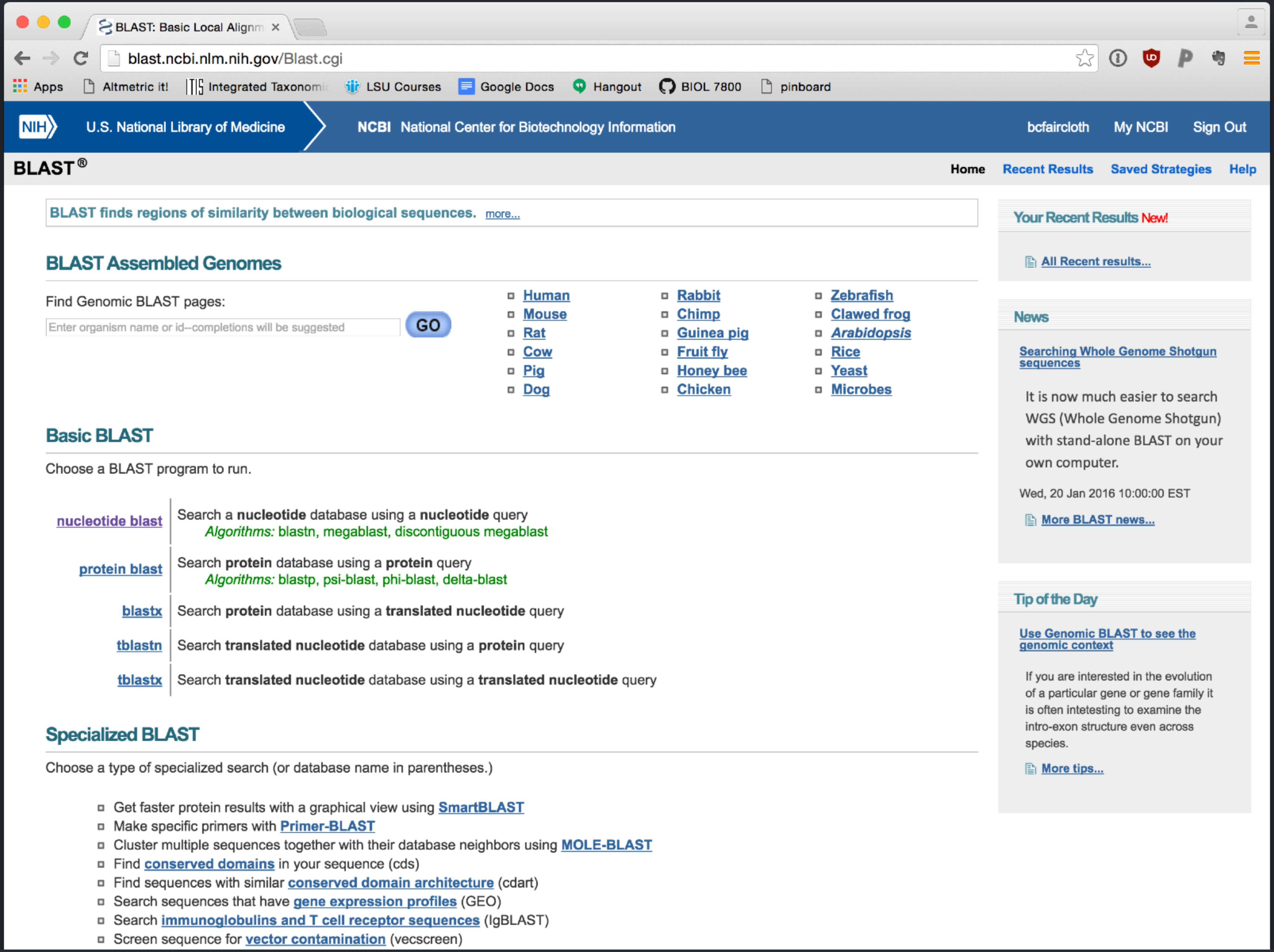
Genome Information

Go to NCBI genomic BLAST page for Alligator mississippiensis
See the NCBI Genome map in MapViewer

Trace records (raw single-pass reads of DNA sequence)		
Sequencing Center Name		
Record counts per type		
CLONEEND	SHOTGUN	ALL
NISC - NIH Intramural Sequencing Center		
-	39,291	39,291
TIGR - The Institute for Genome Research		
172	-	172
Totals per type		
172	39,291	**39,463**

NCBI | BLAST

Basic **L**ocal **A**lignment **S**earch **T**ool



BLAST®

BLAST finds regions of similarity between biological sequences. more...

BLAST Assembled Genomes

Find Genomic BLAST pages:

Enter organism name or id--completions will be suggested

- Human
- Mouse
- Rat
- Cow
- Pig
- Dog
- Rabbit
- Chimp
- Guinea pig
- Fruit fly
- Honey bee
- Chicken
- Zebrafish
- Clawed frog
- *Arabidopsis*
- Rice
- Yeast
- Microbes

Basic BLAST

Choose a BLAST program to run.

nucleotide blast	Search a **nucleotide** database using a **nucleotide** query *Algorithms:* blastn, megablast, discontiguous megablast
protein blast	Search **protein** database using a **protein** query *Algorithms:* blastp, psi-blast, phi-blast, delta-blast
blastx	Search **protein** database using a **translated nucleotide** query
tblastn	Search **translated nucleotide** database using a **protein** query
tblastx	Search **translated nucleotide** database using a **translated nucleotide** query

Specialized BLAST

Choose a type of specialized search (or database name in parentheses.)

- Get faster protein results with a graphical view using SmartBLAST
- Make specific primers with Primer-BLAST
- Cluster multiple sequences together with their database neighbors using MOLE-BLAST
- Find conserved domains in your sequence (cds)
- Find sequences with similar conserved domain architecture (cdart)
- Search sequences that have gene expression profiles (GEO)
- Search immunoglobulins and T cell receptor sequences (IgBLAST)
- Screen sequence for vector contamination (vecscreen)

Your Recent Results New!

All Recent results...

News

Searching Whole Genome Shotgun sequences

It is now much easier to search WGS (Whole Genome Shotgun) with stand-alone BLAST on your own computer.

Wed, 20 Jan 2016 10:00:00 EST

More BLAST news...

Tip of the Day

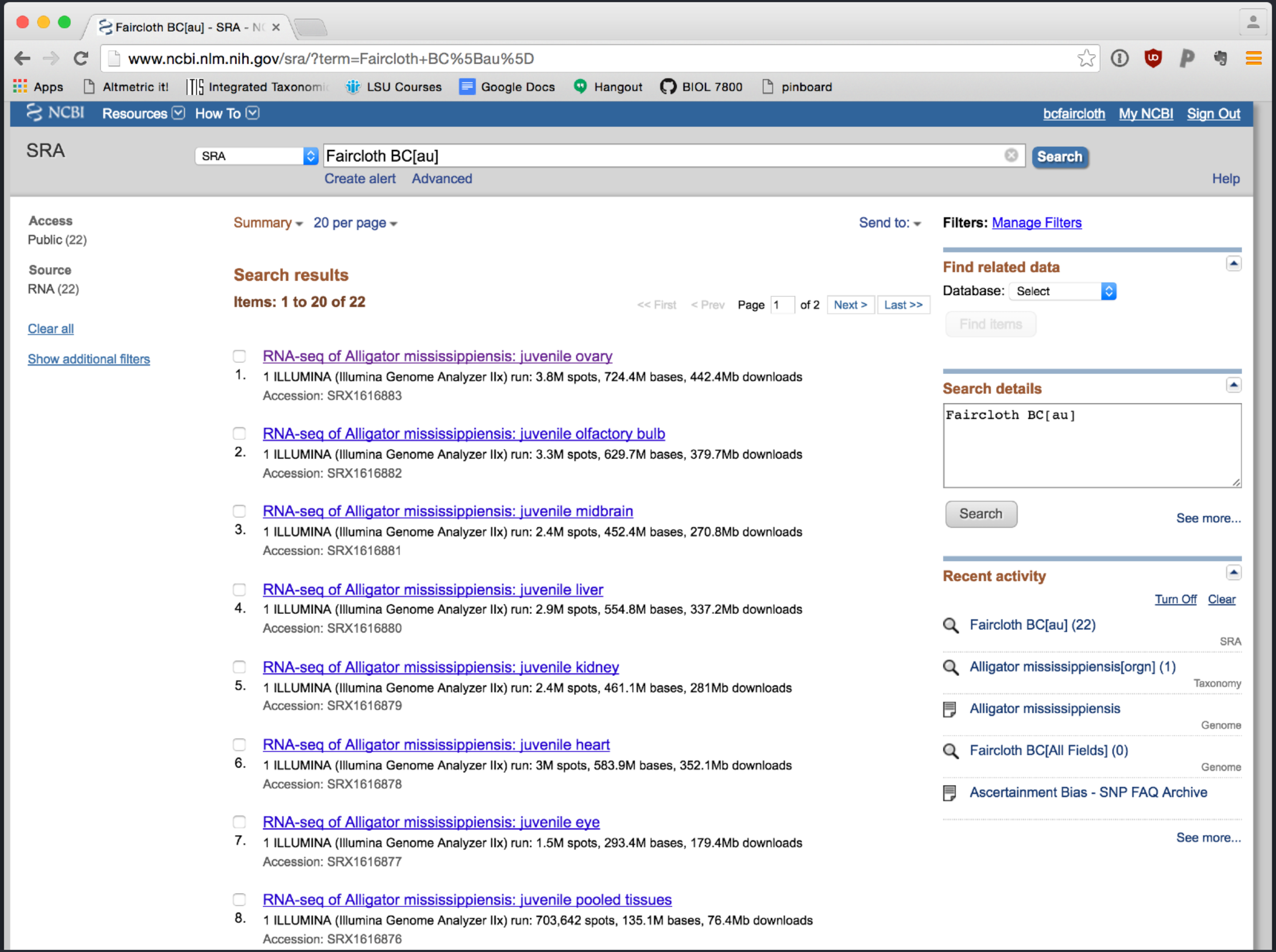
Use Genomic BLAST to see the genomic context

If you are interested in the evolution of a particular gene or gene family it is often intetesting to examine the intro-exon structure even across species.

More tips...

NCBI | SRA

Sequence Read Archive



SRA

SRA: Faircloth BC[au] **Search**

Access
Public (22)

Source
RNA (22)

Search results

Items: 1 to 20 of 22

Page 1 of 2

1. ☐ RNA-seq of Alligator mississippiensis: juvenile ovary
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 3.8M spots, 724.4M bases, 442.4Mb downloads
Accession: SRX1616883

2. ☐ RNA-seq of Alligator mississippiensis: juvenile olfactory bulb
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 3.3M spots, 629.7M bases, 379.7Mb downloads
Accession: SRX1616882

3. ☐ RNA-seq of Alligator mississippiensis: juvenile midbrain
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 2.4M spots, 452.4M bases, 270.8Mb downloads
Accession: SRX1616881

4. ☐ RNA-seq of Alligator mississippiensis: juvenile liver
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 2.9M spots, 554.8M bases, 337.2Mb downloads
Accession: SRX1616880

5. ☐ RNA-seq of Alligator mississippiensis: juvenile kidney
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 2.4M spots, 461.1M bases, 281Mb downloads
Accession: SRX1616879

6. ☐ RNA-seq of Alligator mississippiensis: juvenile heart
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 3M spots, 583.9M bases, 352.1Mb downloads
Accession: SRX1616878

7. ☐ RNA-seq of Alligator mississippiensis: juvenile eye
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 1.5M spots, 293.4M bases, 179.4Mb downloads
Accession: SRX1616877

8. ☐ RNA-seq of Alligator mississippiensis: juvenile pooled tissues
1 ILLUMINA (Illumina Genome Analyzer IIx) run: 703,642 spots, 135.1M bases, 76.4Mb downloads
Accession: SRX1616876

Filters: Manage Filters

Find related data

Database: Select

Search details

Faircloth BC[au]

Search

Recent activity

Faircloth BC[au] (22) — SRA

Alligator mississippiensis[orgn] (1) — Taxonomy

Alligator mississippiensis — Genome

Faircloth BC[All Fields] (0) — Genome

Ascertainment Bias - SNP FAQ Archive

NCBI

National Center for Biotechnology Information

Databases contain an **extraordinary**
amount of (useful!) information

But, interacting with them manually
can be frustrating

(particularly when you have *many* queries)

NCBI

But, interacting with NCBI manually
can be frustrating

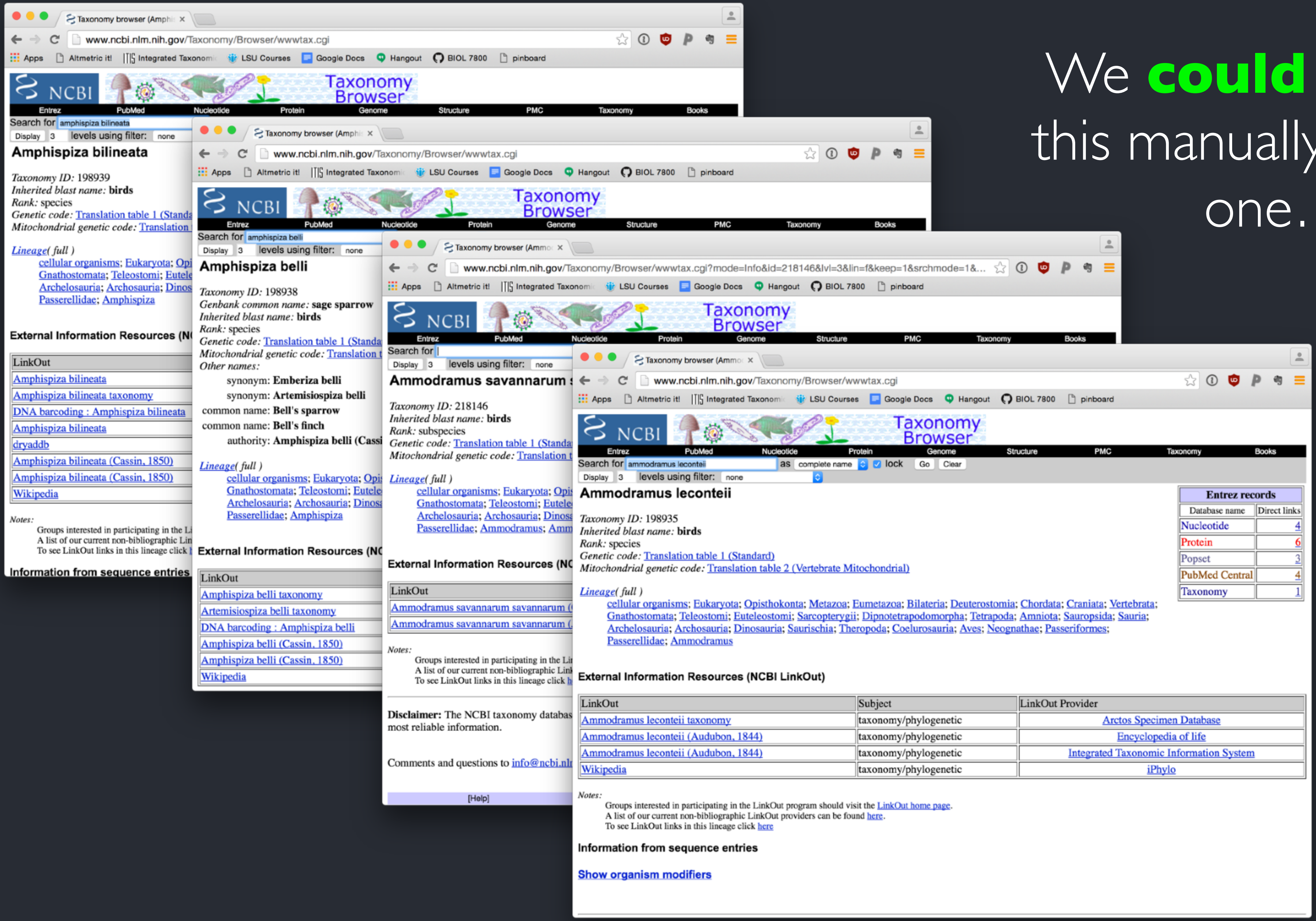
Imagine we have some sparrows….
and we want various information from each

aimophila rufescens
ammodramus leconteii
ammodramus savannarum
amphispiza belli
amphispiza bilineata
arremon aurantiirostris
arremon brunneinucha
arremonops rufivirgatus
atlapetes citrinellus
atlapetes gutturalis
calamospiza melanocorys
chlorospingus ophthalmicus
chondestes grammacus
geothlypis trichas
icterus gularis
junco phaeonotus
melospiza melodia
melozone aberti
melozone leucotis
oriturus superciliosus
passerculus sandwichensis
passerella iliaca
peucaea ruficauda
pezopetes capitalis
pipilo chlorurus
pooecetes gramineus
rhynchospiza strigiceps
spizella arborea
spizella atrogularis
zonotrichia atricapilla

Sparrows

NCBI

Imagine we have some sparrows….



Amphispiza bilineata

Taxonomy ID: 198939
Inherited blast name: **birds**
Rank: species

Amphispiza belli

Taxonomy ID: 198938
Genbank common name: **sage sparrow**
Inherited blast name: **birds**
Rank: species
Other names:
synonym: **Emberiza belli**
synonym: **Artemisiospiza belli**
common name: **Bell's sparrow**
common name: **Bell's finch**

Ammodramus savannarum

Taxonomy ID: 218146
Inherited blast name: **birds**
Rank: subspecies

www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi

Search for: ammodramus leconteii as complete name ☑ lock

Display 3 levels using filter: none

Ammodramus leconteii

Taxonomy ID: 198935
Inherited blast name: **birds**
Rank: species
Genetic code: Translation table 1 (Standard)
Mitochondrial genetic code: Translation table 2 (Vertebrate Mitochondrial)

Lineage(full)
cellular organisms; Eukaryota; Opisthokonta; Metazoa; Eumetazoa; Bilateria; Deuterostomia; Chordata; Craniata; Vertebrata; Gnathostomata; Teleostomi; Euteleostomi; Sarcopterygii; Dipnotetrapodomorpha; Tetrapoda; Amniota; Sauropsida; Sauria; Archelosauria; Archosauria; Dinosauria; Saurischia; Theropoda; Coelurosauria; Aves; Neognathae; Passeriformes; Passerellidae; Ammodramus

Entrez records	
Database name	Direct links
Nucleotide	4
Protein	6
Popset	3
PubMed Central	4
Taxonomy	1

External Information Resources (NCBI LinkOut)

LinkOut	Subject	LinkOut Provider
Ammodramus leconteii taxonomy	taxonomy/phylogenetic	Arctos Specimen Database
Ammodramus leconteii (Audubon, 1844)	taxonomy/phylogenetic	Encyclopedia of life
Ammodramus leconteii (Audubon, 1844)	taxonomy/phylogenetic	Integrated Taxonomic Information System
Wikipedia	taxonomy/phylogenetic	iPhylo

Notes:
Groups interested in participating in the LinkOut program should visit the LinkOut home page.
A list of our current non-bibliographic LinkOut providers can be found here.
To see LinkOut links in this lineage click here

Information from sequence entries

Show organism modifiers

We **could** do all of this manually, one-by-one…

slow,
slow,
slow,
slow,
slow,
slow…

NCBI

Imagine we have some sparrows….

Or, we **could** do all of
this programmatically!

```
% python get_taxonomy.py —tax_list taxonomy.txt
```

```
aimophila rufescens, superkingdom: Eukaryota; kingdom: Metazoa; phylum: Chordata;
subphylum: Craniata; class: Aves; superorder: Neognathae; order: Passeriformes;
family: Passerellidae; genus: Aimophila

ammodramus leconteii, superkingdom: Eukaryota; kingdom: Metazoa; phylum: Chordata;
subphylum: Craniata; class: Aves; superorder: Neognathae; order: Passeriformes;
family: Passerellidae; genus: Ammodramus

ammodramus savannarum, superkingdom: Eukaryota; kingdom: Metazoa; phylum: Chordata;
subphylum: Craniata; class: Aves; superorder: Neognathae; order: Passeriformes;
family: Passerellidae; genus: Ammodramus

amphispiza belli, superkingdom: Eukaryota; kingdom: Metazoa; phylum: Chordata;
subphylum: Craniata; class: Aves; superorder: Neognathae; order: Passeriformes;
family: Passerellidae; genus: Amphispiza
```

Bio.Entrez

The BioPython module for interacting with NCBI

How it works:

1. You determine database and query
2. You send data to NCBI (over internet) using `Bio.Entrez`
3. Your query is processed by their servers
4. The result is returned as XML
5. You or `Bio.Entrez` parse the XML result
6. # do stuff

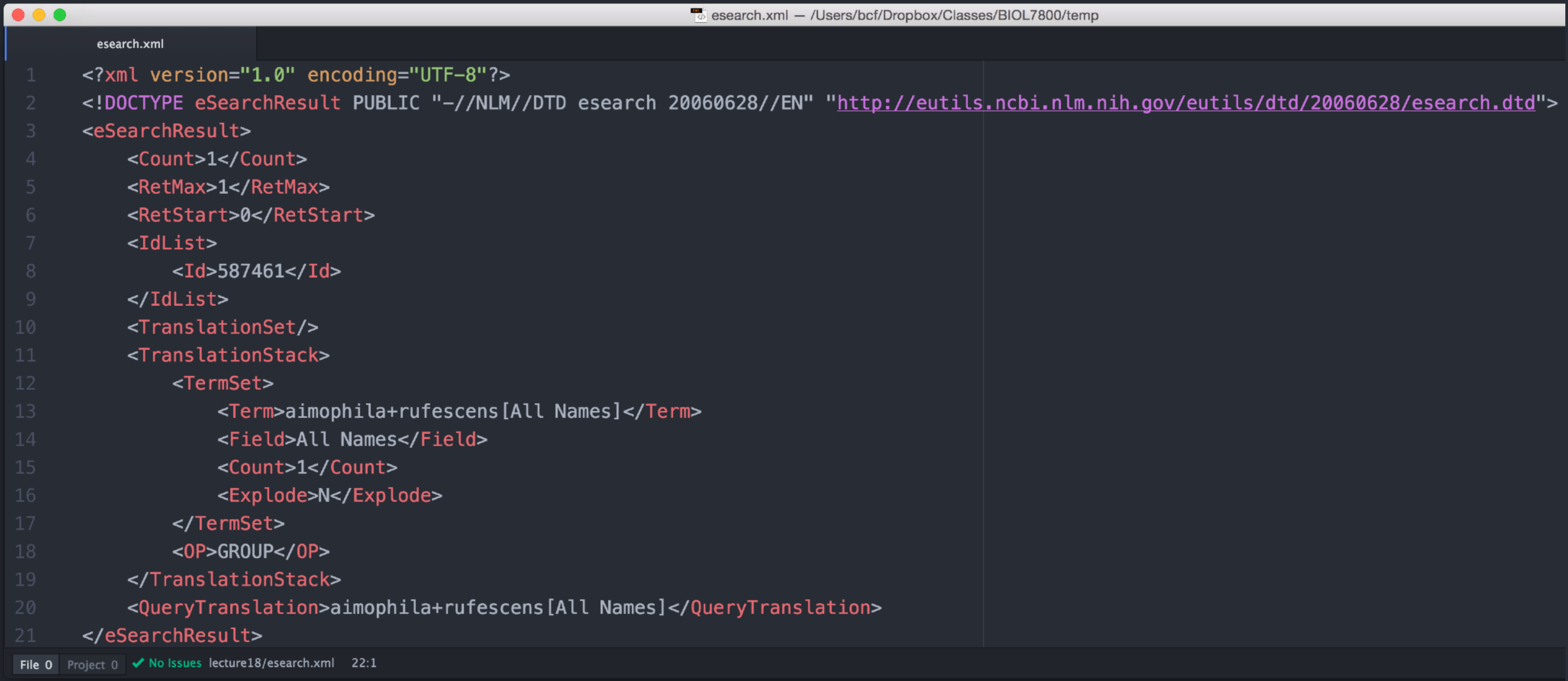
Bio.Entrez

Returns XML

eXtensible Markup Language

Structured Language (like html) for tagging and conveying information

Generally speaking, let BioPython parse this for you…



```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eSearchResult PUBLIC "-//NLM//DTD esearch 20060628//EN" "http://eutils.ncbi.nlm.nih.gov/eutils/dtd/20060628/esearch.dtd">
<eSearchResult>
    <Count>1</Count>
    <RetMax>1</RetMax>
    <RetStart>0</RetStart>
    <IdList>
        <Id>587461</Id>
    </IdList>
    <TranslationSet/>
    <TranslationStack>
        <TermSet>
            <Term>aimophila+rufescens[All Names]</Term>
            <Field>All Names</Field>
            <Count>1</Count>
            <Explode>N</Explode>
        </TermSet>
        <OP>GROUP</OP>
    </TranslationStack>
    <QueryTranslation>aimophila+rufescens[All Names]</QueryTranslation>
</eSearchResult>
```

Bio.Entrez

The BioPython module for interacting with NCBI

Bio.Entrez, which uses NCBI eUtils has **some caveats**:

1. You **must** pass your valid email address
2. You **must** limit queries over 100 items

9:00 PM to 5:00 AM

3. **Not more than** 3 requests per second

Bio.Entrez

Has a number of sub-modules

`Bio.Entrez.einfo()`
Gets info on
Entrez databases

`Bio.Entrez.esearch()`
Searches for Entrez
Identification Values

`Bio.Entrez.epost()`
Searches for Entrez
Identification Values
(in bulk)

`Bio.Entrez.efetch()`
Fetches and returns
Entrez data

And, you need to be aware that **all** items in Entrez databases are retrieved using some form of unique identifier

(and, you have to discover that identifier before you retrieve a record)

Bio.Entrez

Normal Workflow

`Bio.Entrez.einfo()`
Gets info on
Entrez databases

Determine your database
(not usually necessary)

`Bio.Entrez.esearch()`
Searches for Entrez
Identification Values

Fetch unique IDs

↓

`Bio.Entrez.efetch()`
Fetches and returns
Entrez data

Fetch records based on IDs

Bio.Entrez.einfo()

```
In:   from Bio import Entrez
```

Import **Bio.Entrez**

```
In:   Entrez.email = "brant@lsu.edu"
```

Set email

```
In:   entrez_db = Entrez.einfo()
```

Run **einfo()**

```
In:   print(entrez_db.read())
```

Print **einfo()** result in XML

```
Out:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE eInfoResult PUBLIC "-//NLM//DTD einfo 20130322//EN" "http://eutils.ncbi.nlm.nih.gov/eutils/dtd/20130322/einfo.dtd">
<eInfoResult>
<DbList>

        <DbName>pubmed</DbName>
        <DbName>protein</DbName>
        <DbName>nuccore</DbName>
        <DbName>nucleotide</DbName>
        <DbName>nucgss</DbName>
        <DbName>nucest</DbName>
        <DbName>structure</DbName>
        <DbName>genome</DbName>
        <DbName>annotinfo</DbName>
        <DbName>assembly</DbName>
        <DbName>bioproject</DbName>
        <DbName>biosample</DbName>
        <DbName>blastdbinfo</DbName>
        <DbName>gapplus</DbName>
        (...truncated...)
</DbList>
</eInfoResult>
```

All current databases

(truncated)

Bio.Entrez.esearch()

Imagine we want to look up all PubMed Papers
having to do with Eastern Bluebirds (Sialia sialis)

1. We need to get **PubMed Identifiers**…


Sialia sialis

Bio.Entrez.esearch()

Imagine we want to look up all PubMed Papers
having to do with Eastern Bluebirds (Sialia sialis)

1. We need to get PubMed Identifiers…

```
In:   from Bio import Entrez
In:   Entrez.email = "brant@lsu.edu"
In:   esearch_query = Entrez.esearch(db="pubmed", term="eastern bluebird", retmode="xml")
```

database

search terms

return data mode

```
In:   esearch_result = Entrez.read(esearch_query)
In:   print(esearch_result)
```

parse XML with
Entrez.read()

Sialia sialis

Bio.Entrez.esearch()

Imagine we want to look up all PubMed Papers
having to do with Eastern Bluebirds (Sialia sialis)

1. We need to get PubMed Identifiers…

```
In:   from Bio import Entrez
In:   Entrez.email = "brant@lsu.edu"
In:   esearch_query = Entrez.esearch(db="pubmed", term="eastern bluebird", retmode="xml")
```

database

search terms

return data mode

```
In:   esearch_result = Entrez.read(esearch_query)
In:   print(esearch_result)
```

```
Out:{'TranslationStack': [{'Count': '25', 'Explode': 'N', 'Field': 'All Fields',
'Term': 'eastern bluebird[All Fields]'}, 'GROUP'], 'RetStart': '0',
'QueryTranslation': 'eastern bluebird[All Fields]', 'IdList': ['26137900',
'26121017', '24688852', '24457931', '24287858', '24275608', '23524000',
'23432192', '22622467', '22392542', '20514141', '20161226', '19894014',
'19706928', '23538326', '19158016', '19655039', '16764468', '16343620',
'16317482'], 'Count': '25', 'ErrorList': {'FieldNotFound': ['Orgn'],
'PhraseNotFound': []}, 'RetMax': '20', 'TranslationSet': []}
```

Sialia sialis

Bio.Entrez.efetch()

Imagine we want to look up all PubMed Papers
having to do with Eastern Bluebirds (Sialia sialis)

2. Now, need to fetch records from `PubMed`…

Bio.Entrez.efetch()

2. We need to fetch records from PubMed…

```
In:   esearch_result = Entrez.read(esearch_query)

In:   for iden in esearch_result['IdList']:
          pubmed_entry = Entrez.efetch(db="pubmed", id=iden, retmode="xml")
```

↑ database

↑ identifier

↑ return data mode

Sialia sialis

Bio.Entrez.efetch()

2. We need to fetch records from PubMed…

```
In:   esearch_result = Entrez.read(esearch_query)

In:   for iden in esearch_result['IdList'][-1]:
          pubmed_entry = Entrez.efetch(db="pubmed", id=iden, retmode="xml")
          print(Entrez.read(pubmed_entry))
```

```
Out: [{'MedlineCitation': DictElement({'PMID': StringElement('16317482', attributes={'Version': '1'}), 'GeneralNote': [], 'CitationSubset': ['IM'], 'OtherID': [], 'DateRevised':
{'Day': '21', 'Year': '2013', 'Month': '11'}, 'DateCompleted': {'Day': '07', 'Year': '2006', 'Month': '06'}, 'Article': DictElement({'ArticleDate': [DictElement({'Day': '30', 'Year': '2005',
'Month': '11'}, attributes={'DateType': 'Electronic'})], 'Journal': {'JournalIssue': DictElement({'Volume': '15', 'PubDate': {'Year': '2006', 'Month': 'Feb'}, 'Issue': '1'},
attributes={'CitedMedium': 'Print'}), 'ISOAbbreviation': 'Ecotoxicology', 'Title': 'Ecotoxicology (London, England)', 'ISSN': StringElement('0963-9292', attributes={'IssnType':
'Print'})}, 'Language': ['eng'], 'AuthorList': ListElement([DictElement({'Identifier': [], 'Initials': 'DS', 'LastName': 'Henshel', 'AffiliationInfo': [{'Identifier': [], 'Affiliation': 'School of
Public and Environmental Affairs, Indiana University, 1315 E. 10th #340, Bloomington, IN 47401, USA. dhenshel@indiana.edu'}], 'ForeName': 'Diane S'}, attributes={'ValidYN':
'Y'}), DictElement({'Identifier': [], 'Initials': 'DW', 'LastName': 'Sparks', 'AffiliationInfo': [], 'ForeName': 'Daniel W'}, attributes={'ValidYN': 'Y'})], attributes={'CompleteYN': 'Y',
'Type': 'authors'}), 'Pagination': {'MedlinePgn': '9-18'}, 'ArticleTitle': 'Site specific PCB-correlated interspecies differences in organ somatic indices.', 'ELocationID': [],
'PublicationTypeList': [StringElement('Comparative Study', attributes={'UI': 'D003160'}), StringElement('Journal Article', attributes={'UI': 'D016428'}),
StringElement("Research Support, U.S. Gov't, Non-P.H.S.", attributes={'UI': 'D013486'})], 'Abstract': {'AbstractText': ['We correlated site specific differences in the organ ...
truncated...']}}, attributes={'PubModel': 'Print-Electronic'}), 'OtherAbstract': [], 'KeywordList': [], 'DateCreated': {'Day': '10', 'Year': '2006', 'Month': '03'}, 'MedlineJournalInfo':
{'MedlineTA': 'Ecotoxicology', 'NlmUniqueID': '9885956', 'Country': 'United States', 'ISSNLinking': '0963-9292'}, 'ChemicalList': [{'NameOfSubstance':
StringElement('Environmental Pollutants', attributes={'UI': 'D004785'}), 'RegistryNumber': '0'}, {'NameOfSubstance': StringElement('Polychlorinated Biphenyls',
attributes={'UI': 'D011078'}), 'RegistryNumber': 'DFC2HB4I0K'}], 'SpaceFlightMission': [], 'MeshHeadingList': [{'DescriptorName': StringElement('Animals', attributes={'UI':
'D000818', 'MajorTopicYN': 'N'}), 'QualifierName': []}, {'DescriptorName': StringElement('Environmental Monitoring', attributes={'UI': 'D004784', 'MajorTopicYN': 'N'}),
'QualifierName': []}, {'DescriptorName': StringElement('Environmental Pollutants', attributes={'UI': 'D004785', 'MajorTopicYN': 'N'}), 'QualifierName': [StringElement('analysis',
attributes={'UI': 'Q000032', 'MajorTopicYN': 'N'}), StringElement('metabolism', attributes={'UI': 'Q000378', 'MajorTopicYN': 'Y'}), StringElement('toxicity', attributes={'UI':
'Q000633', 'MajorTopicYN': 'N'})]}, {'DescriptorName': StringElement('Passeriformes', attributes={'UI': 'D046109', 'MajorTopicYN': 'N'}), 'QualifierName':
[StringElement('metabolism', attributes={'UI': 'Q000378', 'MajorTopicYN': 'Y'})]}, {'DescriptorName': StringElement('Polychlorinated Biphenyls', attributes={'UI': 'D011078',
'MajorTopicYN': 'N'}), 'QualifierName': [StringElement('analysis', attributes={'UI': 'Q000032', 'MajorTopicYN': 'N'}), StringElement('metabolism', attributes={'UI': 'Q000378',
'MajorTopicYN': 'Y'}), StringElement('toxicity', attributes={'UI': 'Q000633', 'MajorTopicYN': 'N'})]}, {'DescriptorName': StringElement('Species Specificity', attributes={'UI':
'D013045', 'MajorTopicYN': 'N'}), 'QualifierName': []}, {'DescriptorName': StringElement('Tissue Distribution', attributes={'UI': 'D014018', 'MajorTopicYN': 'N'}), 'QualifierName':
[]}]}, attributes={'Owner': 'NLM', 'Status': 'MEDLINE'}), 'PubmedData': {'ArticleIdList': [StringElement('10.1007/s10646-005-0038-0', attributes={'IdType': 'doi'}),
StringElement('16317482', attributes={'IdType': 'pubmed'})], 'PublicationStatus': 'ppublish', 'History': [DictElement({'Day': '2', 'Year': '2005', 'Month': '4'}, attributes={'PubStatus':
'accepted'}), DictElement({'Day': '30', 'Year': '2005', 'Month': '11'}, attributes={'PubStatus': 'aheadofprint'}), DictElement({'Day': '1', 'Year': '2005', 'Minute': '0', 'Month': '12', 'Hour':
'9'}, attributes={'PubStatus': 'pubmed'}), DictElement({'Day': '8', 'Year': '2006', 'Minute': '0', 'Month': '6', 'Hour': '9'}, attributes={'PubStatus': 'medline'}), DictElement({'Day': '1',
'Year': '2005', 'Minute': '0', 'Month': '12', 'Hour': '9'}, attributes={'PubStatus': 'entrez'})]}}]
```

Results are highly nested…

In a list (only 1 element)

Nested Dictionaries

```
[
  {
    'MedlineCitation':
      DictElement({'PMID': StringElement('16317482', attributes={'Version': '1'}),
      'GeneralNote': [],
      'CitationSubset': ['IM'],
      'OtherID': [],
      'DateRevised': {'Day': '21', 'Year': '2013', 'Month': '11'},
      'DateCompleted': {'Day': '07', 'Year': '2006', 'Month': '06'},
       'Article':
        DictElement(
          {'ArticleDate': [DictElement({'Day': '30', 'Year': '2005', 'Month': '11'}, attributes={'DateType': 'Electronic'})],
          'Journal': {
            'JournalIssue':
              DictElement(
                {'Volume': '15', 'PubDate': {'Year': '2006', 'Month': 'Feb'}, 'Issue': '1'}, attributes={'CitedMedium': 'Print'}),
            'ISOAbbreviation': 'Ecotoxicology',
            'Title': 'Ecotoxicology (London, England)',
            'ISSN': StringElement('0963-9292', attributes={'IssnType': 'Print'})},
            'Language': ['eng'],
            'AuthorList':
              ListElement([
                DictElement({'Identifier': [], 'Initials': 'DS', 'LastName': 'Henshel', 'AffiliationInfo': [{'Identifier': [], 'Affiliation': 'School
                  of Public and Environmental Affairs, Indiana University, 1315 E. 10th #340, Bloomington, IN 47401, USA.
                  dhenshel@indiana.edu'}], 'ForeName': 'Diane S'}, attributes={'ValidYN': 'Y'}),
                DictElement({'Identifier': [], 'Initials': 'DW', 'LastName': 'Sparks', 'AffiliationInfo': [], 'ForeName': 'Daniel W'},
                  attributes={'ValidYN': 'Y'})],
              attributes={'CompleteYN': 'Y', 'Type': 'authors'}),
              'Pagination': {'MedlinePgn': '9-18'},
              'ArticleTitle': 'Site specific PCB-correlated interspecies differences in organ somatic indices.',
              'ELocationID': [],
              (...TRUNCATED...)
    'PubmedData':
      {'ArticleIdList': [StringElement('10.1007/s10646-005-0038-0', attributes={'IdType': 'doi'}), StringElement('16317482',
attributes={'IdType': 'pubmed'})],
      'PublicationStatus': 'ppublish',
      (...TRUNCATED...)
  }
]
```

Bio.Entrez.efetch()

2. We need to fetch records from PubMed…

Results are highly nested…

```
In:   esearch_result = Entrez.read(esearch_query)

In:   for iden in esearch_result['IdList'][-1]:
          pubmed_entry = Entrez.efetch(db="pubmed", id=iden, retmode="xml")
          result = Entrez.read(pubmed_entry)
          break

In:   result.keys()
Out:dict_keys(['PubmedArticle', 'PubmedBookArticle'])

In:   result['PubmedArticle'][0].keys()
Out:dict_keys(['MedlineCitation', 'PubmedData'])

In:   result['PubmedArticle'][0]['MedlineCitation']
Out: DictElement({'GeneralNote': [], 'OtherAbstract': [], 'KeywordList':
[ListElement([StringElement("Swainson's thrush",
attributes={'MajorTopicYN': 'N'}), StringElement('fat stores',
attributes={'MajorTopicYN': 'N'}),  (...TRUNCATED...)
```

Sialia sialis

Bio.Entrez.efetch()

2. We need to fetch records from PubMed…

Results are highly nested…

```
In:  result['PubmedArticle'][0]['MedlineCitation']['Article']['AuthorList']

Out: ListElement([DictElement({'AffiliationInfo': [DictElement({'Identifier': [],
'Affiliation': 'Department of Natural Resources and Environmental Sciences,
University of Illinois, Urbana-Champaign, IL, USA mpward@illinois.edu.'},
attributes={}), DictElement({'Identifier': [], 'Affiliation': 'Illinois Natural History
Survey, Prairie Research Institute, University of Illinois, Urbana-Champaign,
IL, USA.'}, attributes={})], 'Identifier': [StringElement('http://orcid.org/
0000-0002-1081-6244', attributes={'Source': 'ORCID'})], 'LastName': 'Ward',
'ForeName': 'Michael P', 'Initials': 'MP'}, attributes={'ValidYN': 'Y'}),
(...TRUNCATED...)
```

Sialia sialis

Now imagine we want to look up all Sequence Data
having to do with Eastern Bluebirds (Sialia sialis)

How do we do this?

What are the steps?

Sialia sialis

Bio.Entrez.esearch()

Imagine we want to look up all Sequence Data
having to do with Eastern Bluebirds (Sialia sialis)

1. We need to get Taxonomy Identifiers…

```
In:  from Bio import Entrez
In:  Entrez.email = "brant@lsu.edu"
In:  esearch_query = Entrez.esearch(db="taxonomy", term="eastern bluebird", retmode="xml")
```

database

search terms

return data mode

```
In:  esearch_result = Entrez.read(esearch_query)
In:  print(esearch_result)

Out:{'TranslationStack': [{'Count': '1', 'Explode': 'N', 'Field': 'All Names',
'Term': 'eastern bluebird[All Names]'}, 'GROUP'], 'RetStart': '0',
'QueryTranslation': 'eastern bluebird[All Names]', 'IdList': ['172413'],
'Count': '1', 'RetMax': '1', 'TranslationSet': []}
```

What's important in the above?

Sialia sialis

Bio.Entrez.esearch()

Imagine we want to look up all Sequence Data
having to do with Eastern Bluebirds (Sialia sialis)

1. We need to get Taxonomy Identifiers…

```
In:  from Bio import Entrez
In:  Entrez.email = "brant@lsu.edu"
In:  esearch_query = Entrez.esearch(db="taxonomy", term="eastern bluebird", retmode="xml")
```

database search terms return data mode

```
In:  esearch_result = Entrez.read(esearch_query)
In:  print(esearch_result)
```

```
Out:{'TranslationStack': [{'Count': '1', 'Explode': 'N', 'Field': 'All Names',
'Term': 'eastern bluebird[All Names]'}, 'GROUP'], 'RetStart': '0',
'QueryTranslation': 'eastern bluebird[All Names]', 'IdList': ['172413'],
'Count': '1', 'RetMax': '1', 'TranslationSet': []}
```

What's important in the above?

Sialia sialis

Bio.Entrez.esearch()

Imagine we want to look up all Sequence Data
having to do with Eastern Bluebirds (Sialia sialis)

2. We need to get **Genbank Identifiers**…

```
In:   bb_id = esearch_result['IdList']
In:   bb_id
Out:['172413']
In:   bb_id = "txid{}".format(bb_id[0])
In:   genbank_entries = Entrez.esearch(db="nucleotide", term=bb_id, retmode="xml")
```

database

search terms

return data mode

```
In:   esearch_result = Entrez.read(genbank_entries)
In:   print(esearch_result)
```

```
Out: {'TranslationStack': [{'Count': '212', 'Explode': 'N', 'Field': 'All Fields', 'Term': 'txid172413[All
Fields]'}, 'GROUP'], 'RetStart': '0', 'QueryTranslation': 'txid172413[All Fields]', 'IdList':
['1002002998', '295001782', '514252612', '514252610', '514252608', '514252606',
'514252604', '514252602', '514252600', '514252598', '514252596', '514252594', '514252592',
'514252590', '514252588', '442773967', '442773966', '442773885', '442773884',
'442773883'], 'Count': '212', 'RetMax': '20', 'TranslationSet': []}
```

Sialia sialis

Bio.Entrez.efetch()

3. We need to fetch records from Genbank…

```
In:   esearch_result = Entrez.read(esearch_query)

In:   bb_seq_id = esearch_result['IdList']
In:   for record in bb_seq_id['IdList']:
          genbank_entry = Entrez.efetch(db="nucleotide", id=record, retmode="xml")
          data = Entrez.read(genbank_entry)
          # do stuff
```

Sialia sialis

Bio.Entrez.efetch()

3. We need to fetch records from Genbank…
Results are highly nested…

```
[{'GBSeq_topology': 'linear', 'GBSeq_locus': 'KT852372', 'GBSeq_feature-table': [{'GBFeature_key': 'source', 'GBFeature_quals':
[{'GBQualifier_value': 'Sialia sialis', 'GBQualifier_name': 'organism'}, {'GBQualifier_value': 'genomic DNA', 'GBQualifier_name': 'mol_type'},
{'GBQualifier_value': 'taxon:172413', 'GBQualifier_name': 'db_xref'}, {'GBQualifier_value': 'posterior telencephalon', 'GBQualifier_name':
'tissue_type'}, {'GBQualifier_value': 'nestling', 'GBQualifier_name': 'dev_stage'}, {'GBQualifier_value': 'USA: Auburn, AL', 'GBQualifier_name':
'country'}, {'GBQualifier_value': 'Apr-2011', 'GBQualifier_name': 'collection_date'}], 'GBFeature_location': '1..656', 'GBFeature_intervals':
[{'GBInterval_accession': 'KT852372.1', 'GBInterval_to': '656', 'GBInterval_from': '1'}]}, {'GBFeature_partial5': StringElement('',
attributes={'value': 'true'}), 'GBFeature_partial3': StringElement('', attributes={'value': 'true'}), 'GBFeature_key': 'gene', 'GBFeature_quals':
[{'GBQualifier_value': 'ESR1', 'GBQualifier_name': 'gene'}], 'GBFeature_location': 'complement(<1..>656)', 'GBFeature_intervals':
[{'GBInterval_iscomp': StringElement('', attributes={'value': 'true'}), 'GBInterval_accession': 'KT852372.1', 'GBInterval_to': '1',
'GBInterval_from': '656'}]}, {'GBFeature_partial5': StringElement('', attributes={'value': 'true'}), 'GBFeature_key': 'mRNA', 'GBFeature_quals':
[{'GBQualifier_value': 'ESR1', 'GBQualifier_name': 'gene'}, {'GBQualifier_value': 'estrogen receptor 1', 'GBQualifier_name': 'product'},
{'GBQualifier_value':
'GACATATCTGCTTTCACTGTCAACAGTGAGATCGGCAGCTGGCAGTCCACCTAAAAAAATACCAGAACCTATTCTGAGGCTCAGCAGAAATCA
TAATAAAGCAGCACTACACAGGTATTCCTCTAATCTCGGCAAGGAGGAGGATACAAACAAATGCAGAGTGAGCACTGAAAATGCACATTCTCTCT
CTTCTTCCCTCCCTCTCCCTCTCTCTAACTTTTAGGAACTCCTCATCTGCACTAATATGCATTACAAGGTGCTGGCTGCAAGCCAGGAGCCCTTG
CCTTGTGCCGACGTTGGTTTAAACATCACTTCAGAAACAATTAGTTCTGGAGTTATACCAAAGAGTTGTCCGTCTAAATGTTACTTCATTCTTACTG
ATCTTTTAATTTTTCACTTCAGGGTACAGTGCTTTGCAGAAACCAACGGGAAAGAACACTTAAATTCACCCAGGAAGACTAGTCCAGTGGTCAGT
CAGATTACTGCCTTTTATTCCTCAGCAGGCTCCAATCTATCTGTGCTTATCTCTCCTCCTGTTTGATTCATTTGCATGTTTGCTAGAAATATGTAATGT
GTACATTGTACTGACCCTGGGCAGGACTGTGCTGTAACCGAGATAGGGAAAGCAGAATGTGTGTATGCGTGTGTGTGCAAG',
'GBQualifier_name': 'transcription'}], 'GBFeature_location': 'complement(1..>656)', 'GBFeature_intervals': [{'GBInterval_iscomp':
StringElement('', attributes={'value': 'true'}), 'GBInterval_accession': 'KT852372.1', 'GBInterval_to': '1', 'GBInterval_from': '656'}]},
{'GBFeature_partial5': StringElement('', attributes={'value': 'true'}), 'GBFeature_key': "5'UTR", 'GBFeature_quals': [{'GBQualifier_value': 'ESR1',
'GBQualifier_name': 'gene'}], 'GBFeature_location': 'complement(1..>656)', 'GBFeature_intervals': [{'GBInterval_iscomp': StringElement('',
attributes={'value': 'true'}), 'GBInterval_accession': 'KT852372.1', 'GBInterval_to': '1', 'GBInterval_from': '656'}]}], 'GBSeq_primary-accession':
'KT852372', 'GBSeq_length': '656', 'GBSeq_references': [{'GBReference_authors': ['Bentz,A.B.', 'Sirman,A.', 'Wada,H.', 'Navara,K.J.',
'Hood,W.R.'], 'GBReference_title': 'Maternal environment alters epigenetic state of estrogen receptor alpha in wild Eastern Bluebird (Sialia
sialis) nestlings', 'GBReference_journal': 'Unpublished', 'GBReference_reference': '1', 'GBReference_position': '1..656'} ...TRUNCATED...
```

Bio.Entrez.efetch()

3. We need to fetch records from Genbank…
Results are highly nested…

We can drill down into results

```
In:   data = Entrez.read(genbank_entry)
In:   data[0].keys()
Out:dict_keys(['GBSeq_topology', 'GBSeq_locus', 'GBSeq_feature-table', 'GBSeq_primary-
accession', 'GBSeq_length', 'GBSeq_references', 'GBSeq_strandedness', 'GBSeq_comment',
'GBSeq_update-date', 'GBSeq_sequence', 'GBSeq_create-date', 'GBSeq_taxonomy',
'GBSeq_source', 'GBSeq_accession-version', 'GBSeq_other-seqids', 'GBSeq_division',
'GBSeq_organism', 'GBSeq_moltype', 'GBSeq_definition'])
```

```
In:   data[0]['GBSeq_sequence']
'cttgcacacacacgcatacacacattctgctttccctatctcggttacagcacagtcctgcccagggtcagtacaatgtacacattacatatttctagcaaacatgcaaa
tgaatcaaacaggaggagagataagcacagatagattggagcctgctgaggaataaaaggcagtaatctgactgaccactggactagtcttcctgggtgaatttaagtgtt
ctttcccgttggtttctgcaaagcactgtaccctgaagtgaaaaattaaaagatcagtaagaatgaagtaacatttagacggacaactctttggtataactccagaactaa
ttgtttctgaagtgatgtttaaaccaacgtcggcacaaggcaagggctcctggcttgcagccagcaccttgtaatgcatattagtgcagatgaggagttcctaaaagttag
agagagggagagggagggaagaagagagagaatgtgcattttcagtgctcactctgcatttgtttgtatcctcctccttgccgagattagaggaatacctgtgtagtgctg
ctttattatgatttctgctgagcctcagaataggttctggtattttttttaggtggactgccagctgccgatctcactgttgacagtgaaagcagatatgtc'
```

Bio.Entrez.efetch()

3. We need to fetch records from **Genbank**…
Results are **highly nested**…

We can drill down into results

```
In:  data = Entrez.read(genbank_entry)
In:  data[0]['GBSeq_sequence']
'cttgcacacacacgcatacacacattctgctttccctatctcggttacagcacagtcctgcccagggtcagtacaatgtacacattacatatttctagcaaacatgcaaa
tgaatcaaacaggaggagagataagcacagatagattggagcctgctgaggaataaaaggcagtaatctgactgaccactggactagtcttcctgggtgaatttaagtgtt
ctttcccgttggtttctgcaaagcactgtaccctgaagtgaaaaattaaaagatcagtaagaatgaagtaacatttagacggacaactctttggtataactccagaactaa
ttgtttctgaagtgatgtttaaaccaacgtcggcacaaggcaagggctcctggcttgcagccagcaccttgtaatgcatattagtgcagatgaggagttcctaaaagttag
agagagggagagggagggaagaagagagagaatgtgcattttcagtgctcactctgcatttgtttgtatcctcctccttgccgagattagaggaatacctgtgtagtgctg
ctttattatgatttctgctgagcctcagaataggttctggtatttttttaggtggactgccagctgccgatctcactgttgacagtgaaagcagatatgtc'
```

And you could use this information to
make sequence record, perform other analyses, etc

But, there is a smarter/easier way…

Bio.Entrez.efetch()

3. We need to fetch records from **Genbank**…
Results are **highly nested**…

We can change the type and mode of our query!

```
In:   esearch_result = Entrez.read(esearch_query)

In:   bb_seq_id = esearch_result['IdList']
In:   for record in bb_seq_id['IdList']:
          genbank_record = Entrez.efetch(db="nucleotide", id=record, rettype="gb", retmode="text")
```

return data type

return data mode

The smarter/easier way…

Bio.Entrez.efetch()

3. We need to fetch records from **Genbank**…
Results are **highly nested**…

We can change the type and mode of our query!

```
In:   esearch_result = Entrez.read(esearch_query)

In:   bb_seq_id = esearch_result['IdList']
In:   for record in bb_seq_id['IdList']:
          genbank_record = Entrez.efetch(db="nucleotide", id=record, rettype="gb", retmode="text")
          record = SeqIO.read(genbank_record, 'genbank')
```

↑

And, **SeqIO** will parse 'genbank' formats

The smarter/easier way…

Bio.Entrez.efetch()

3. We need to fetch records from Genbank…
Results are highly nested…

We can change the type and mode of our query!

```
In:   esearch_result = Entrez.read(esearch_query)

In:   bb_seq_id = esearch_result['IdList']
In:   for record in bb_seq_id['IdList']:
          genbank_record = Entrez.efetch(db="nucleotide", id=record, rettype="gb", retmode="text"")
          record = SeqIO.read(genbank_record, 'genbank')
          # do stuff
```

↑

So, we can use **SeqIO** to parse the returned record

And then it behaves like a **SequenceRecord**

↓

```
In:   record
Out:SeqRecord(seq=Seq('CTTGCACACACACGCATACACACATTCTGCTTTCCCTATCTCGGTTACAGC
ACA...GTC', IUPACAmbiguousDNA()), id='KT852372.1', name='KT852372', description="Sialia
sialis estrogen receptor 1 (ESR1) gene, 5' UTR.", dbxrefs=[])
```

One last item…

Bio.Entrez

The BioPython module for interacting with NCBI

`Bio.Entrez`, which uses NCBI eUtils has **some caveats**:

1. You **must** pass your valid email address
2. You **must** limit queries over 100 items

`9:00 PM to 5:00 AM`

3. **Not more than** 3 requests per second

One last item…

So, how do you limit the speed of requests?

Using your old friend, the time module

```
In:  for item in range(0, 10, 1):
         print(item)
```

One last item…

So, how do you limit the speed of requests?

Using your old friend, the time module

```
In:  import time
In:  for item in range(0, 10, 1):
         print(item)
         time.sleep(1)
```

time.sleep()
pauses execution

by () <u>seconds</u>