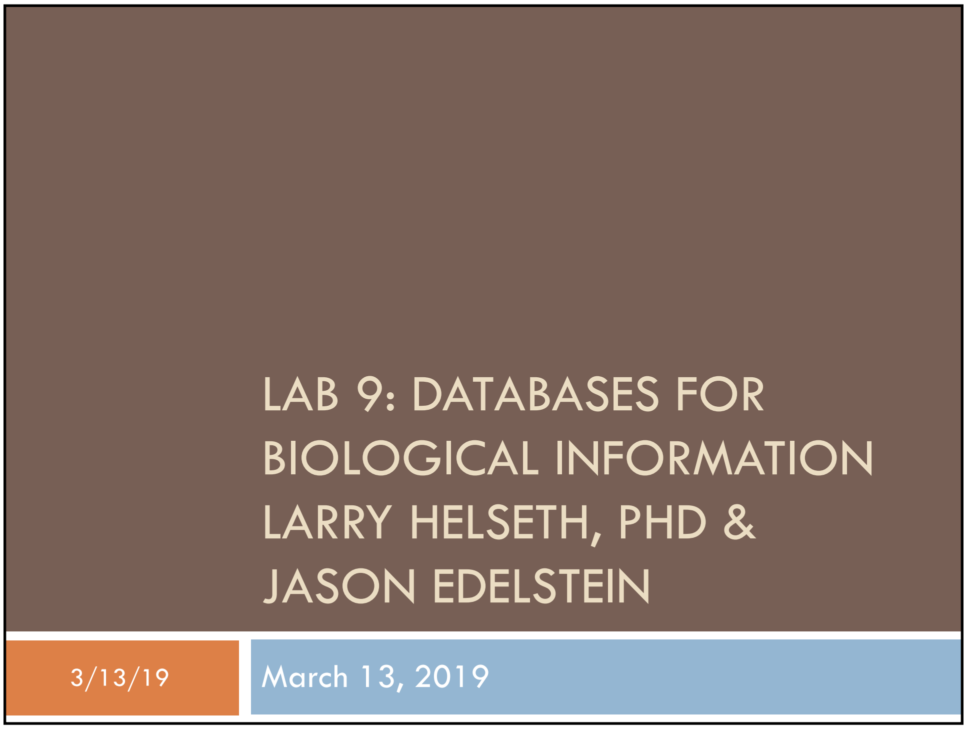
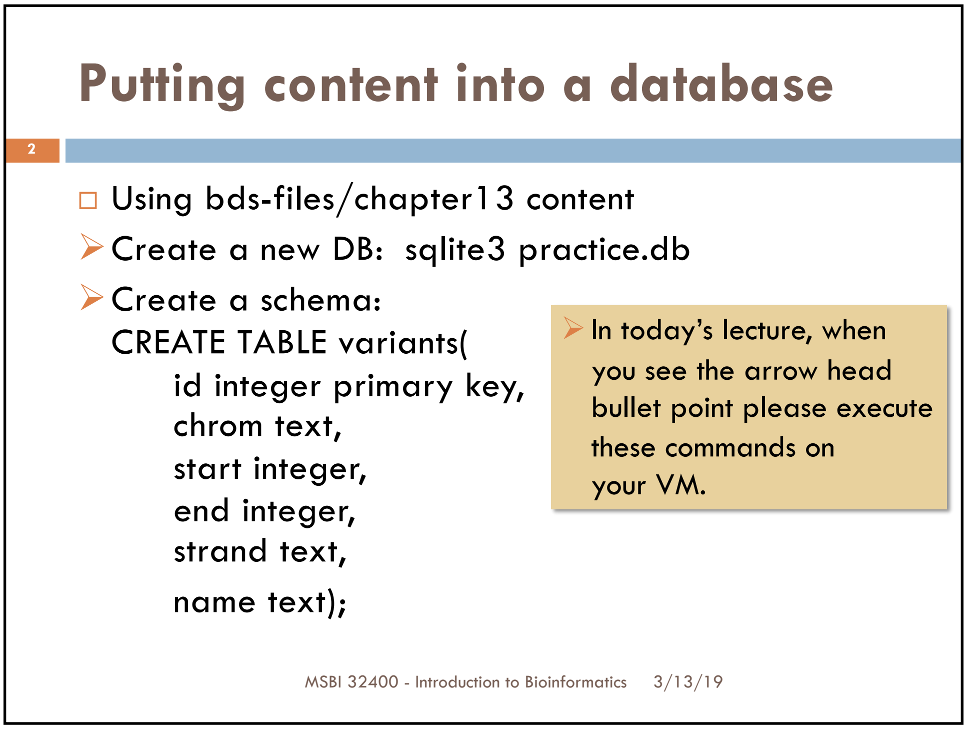


LAB 9: DATABASES FOR BIOLOGICAL INFORMATION LARRY HELSETH, PHD & JASON EDELSTEIN

3/13/19

March 13, 2019



Putting content into a database

2

- □ Using bds-files/chapter13 content
- ➢ Create a new DB: sqlite3 practice.db
- ➢ Create a schema:

```
CREATE TABLE variants(
     id integer primary key,
     chrom text,
     start integer,
     end integer,
     strand text,
     name text);
```

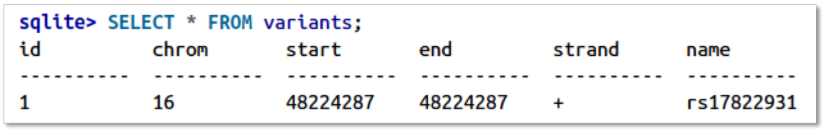
➢ In today's lecture, when you see the arrow head bullet point please execute these commands on your VM.

MSBI 32400 - Introduction to Bioinformatics 3/13/19

Adding content to table

3

- □ Follow syntax like:
  INSERT INTO tablename (column1, column2)
  VALUES (value1, value2);
- INSERT INTO variants(id, chrom, start, end, strand, name)
  VALUES(NULL, "16", 48224287, 48224287, "+", "rs17822931");
  - ***start*** and ***end*** are Integers. Use quotes for strings.
- Confirm insertion by “SELECT * FROM variants;”<enter>:

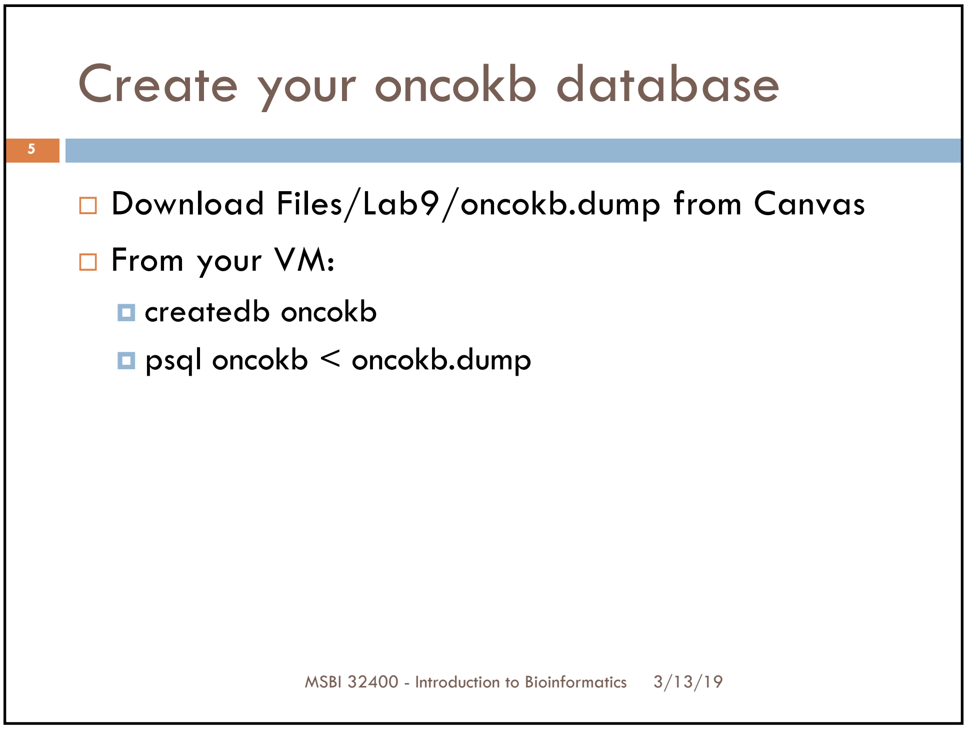


```
sqlite> SELECT * FROM variants;
id          chrom       start       end         strand      name
----------  ----------  ----------  ----------  ----------  ----------
1           16          48224287    48224287    +           rs17822931
```

Let’s install pgIIIAdmin on your VM

4

- Install pg3admin on your VM using:
  - sudo yum install pgadmin3
  - sudo su postgres
    - \password
    - pass
    - exit
  - Start pgAdminIII from menu
  - Click the “plug” to connect then configure by clicking on “Server Groups” and adding msbi32400lab5 and localhost

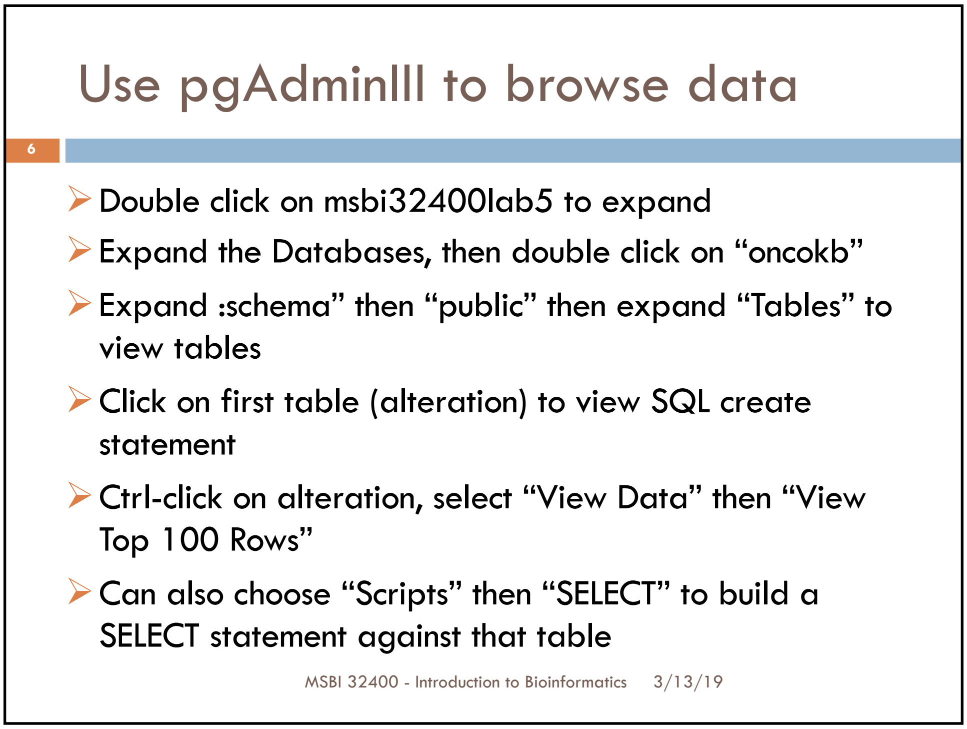


Create your oncokb database

5

- Download Files/Lab9/oncokb.dump from Canvas
- From your VM:
  - createdb oncokb
  - psql oncokb < oncokb.dump

MSBI 32400 - Introduction to Bioinformatics 3/13/19

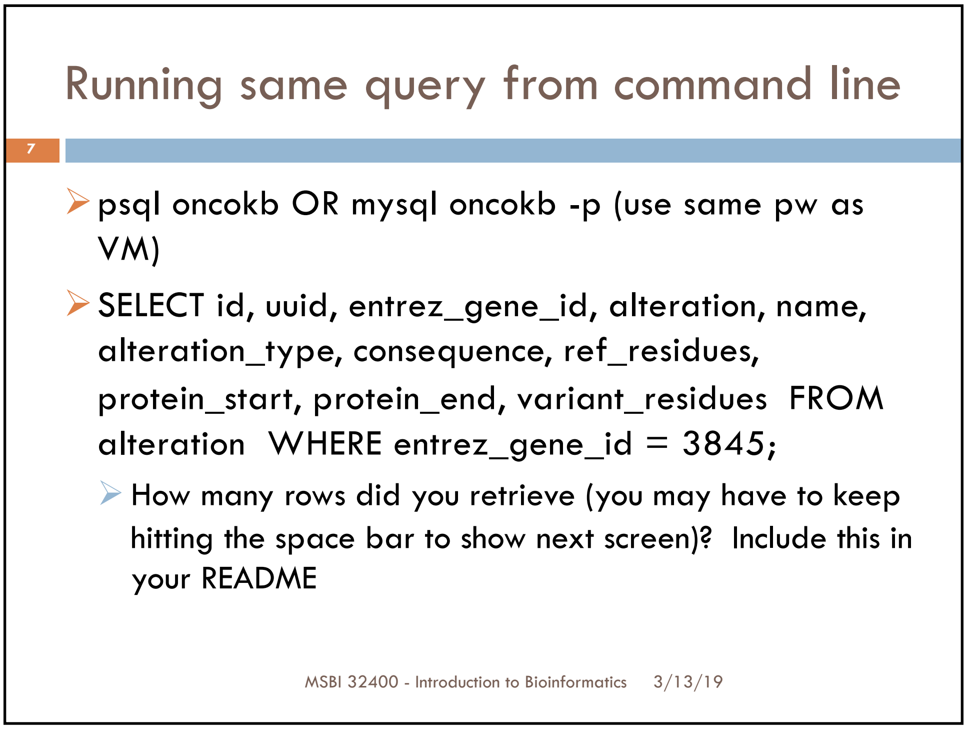


Use pgAdminIII to browse data

6

- Double click on msbi32400lab5 to expand
- Expand the Databases, then double click on “oncokb”
- Expand :schema” then “public” then expand “Tables” to view tables
- Click on first table (alteration) to view SQL create statement
- Ctrl-click on alteration, select “View Data” then “View Top 100 Rows”
- Can also choose “Scripts” then “SELECT” to build a SELECT statement against that table

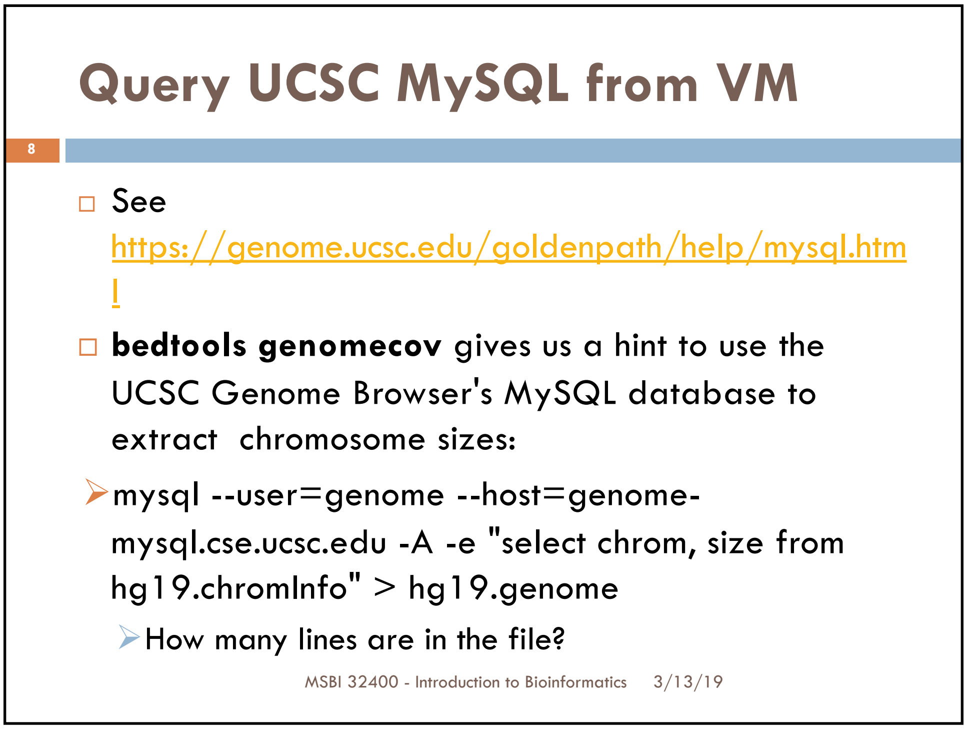
MSBI 32400 - Introduction to Bioinformatics 3/13/19



Running same query from command line

7

- psql oncokb OR mysql oncokb -p (use same pw as VM)
- SELECT id, uuid, entrez_gene_id, alteration, name, alteration_type, consequence, ref_residues, protein_start, protein_end, variant_residues FROM alteration WHERE entrez_gene_id = 3845;
  - How many rows did you retrieve (you may have to keep hitting the space bar to show next screen)? Include this in your README

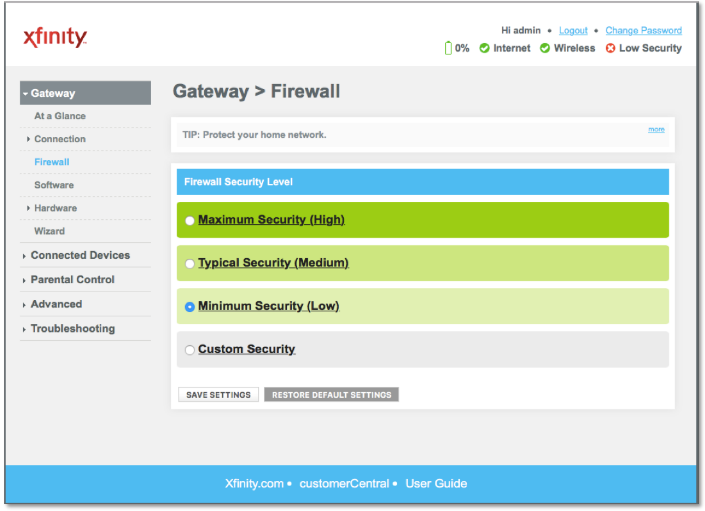


Query UCSC MySQL from VM

8

- □ See https://genome.ucsc.edu/goldenpath/help/mysql.html
- □ **bedtools genomecov** gives us a hint to use the UCSC Genome Browser's MySQL database to extract chromosome sizes:
- mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e "select chrom, size from hg19.chromInfo" > hg19.genome
  - How many lines are in the file?

Problems connecting at home?



Gateway > Firewall

TIP: Protect your home network.

Firewall Security Level

○ Maximum Security (High)

○ Typical Security (Medium)

● Minimum Security (Low)

○ Custom Security

Backup your data with DUMP

- From the command line (not inside psql or mysql) run:
  - pg_dump mydatabase > mydatabase.dump (or .SQL)
  - mysqldump mydatabase > mydatabase.dump (or .SQL)
- Contains SQL to recreate your database and insert data
  - Check help/Google for dumping data only, table only, dump without database owner, etc.
- Common during development to run:
  - pg_dump ***dbname*** > dbname.SQL
  - dropdb ***dbname***
  - createdb ***dbname***
  - psql ***dbname*** < dbname.SQL

Can also use GUI tools

SQLite dump

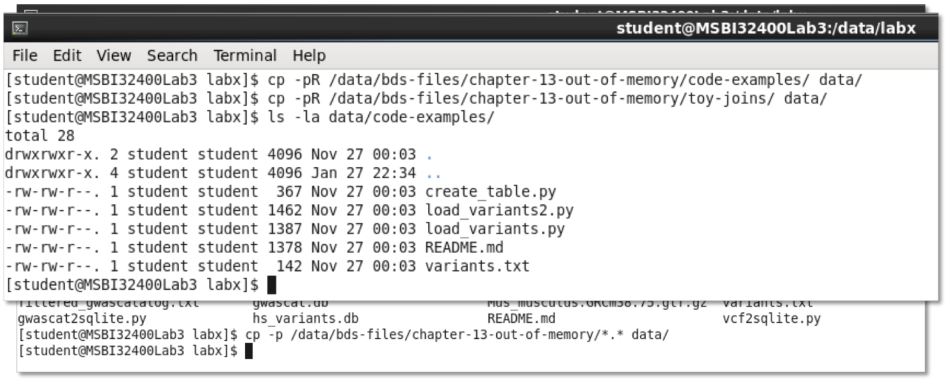
11

- □ Buffalo has an example of using ".dump" in sqlite3 from the end of Chapter 13:
- □ Can use this to backup and duplicate your database:
  - ➢ sqlite3 variants.db ".dump" > dump.sql
  - ➢ sqlite3 variants-duplicate.db < dump.sql

Lab

12

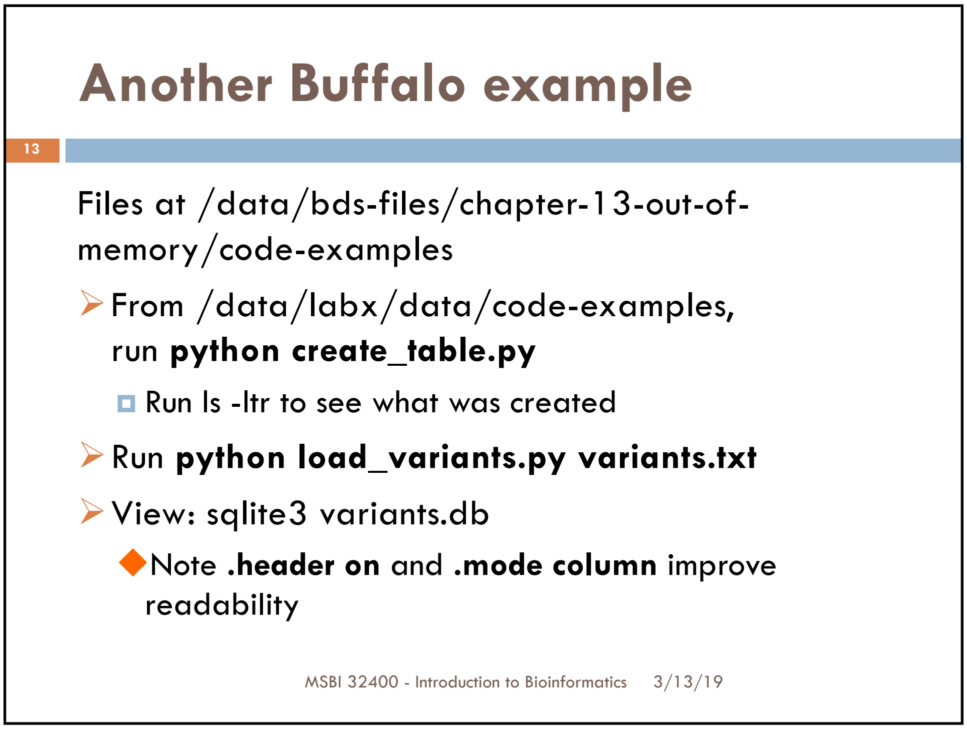
- ➢ Create /data/lab9 then bin, doc, data, src & results
- ➢ Copy files from the /data/bds-files/chapter13-out-of-memory directory to data/lab9/data



student@MSBI32400Lab3:/data/labx
File Edit View Search Terminal Help

```
[student@MSBI32400Lab3 labx]$ cp -pR /data/bds-files/chapter-13-out-of-memory/code-examples/ data/
[student@MSBI32400Lab3 labx]$ cp -pR /data/bds-files/chapter-13-out-of-memory/toy-joins/ data/
[student@MSBI32400Lab3 labx]$ ls -la data/code-examples/
total 28
drwxrwxr-x. 2 student student 4096 Nov 27 00:03 .
drwxrwxr-x. 4 student student 4096 Jan 27 22:34 ..
-rw-rw-r--. 1 student student  367 Nov 27 00:03 create_table.py
-rw-rw-r--. 1 student student 1462 Nov 27 00:03 load_variants2.py
-rw-rw-r--. 1 student student 1387 Nov 27 00:03 load_variants.py
-rw-rw-r--. 1 student student 1378 Nov 27 00:03 README.md
-rw-rw-r--. 1 student student  142 Nov 27 00:03 variants.txt
[student@MSBI32400Lab3 labx]$ 
```

```
filtered_gwascatalog.txt    gwascat.db              Mus_musculus.GRCm38.75.gtf.gz  variants.txt
gwascat2sqlite.py           hs_variants.db          README.md                      vcf2sqlite.py
[student@MSBI32400Lab3 labx]$ cp -p /data/bds-files/chapter-13-out-of-memory/*.* data/
[student@MSBI32400Lab3 labx]$ 
```



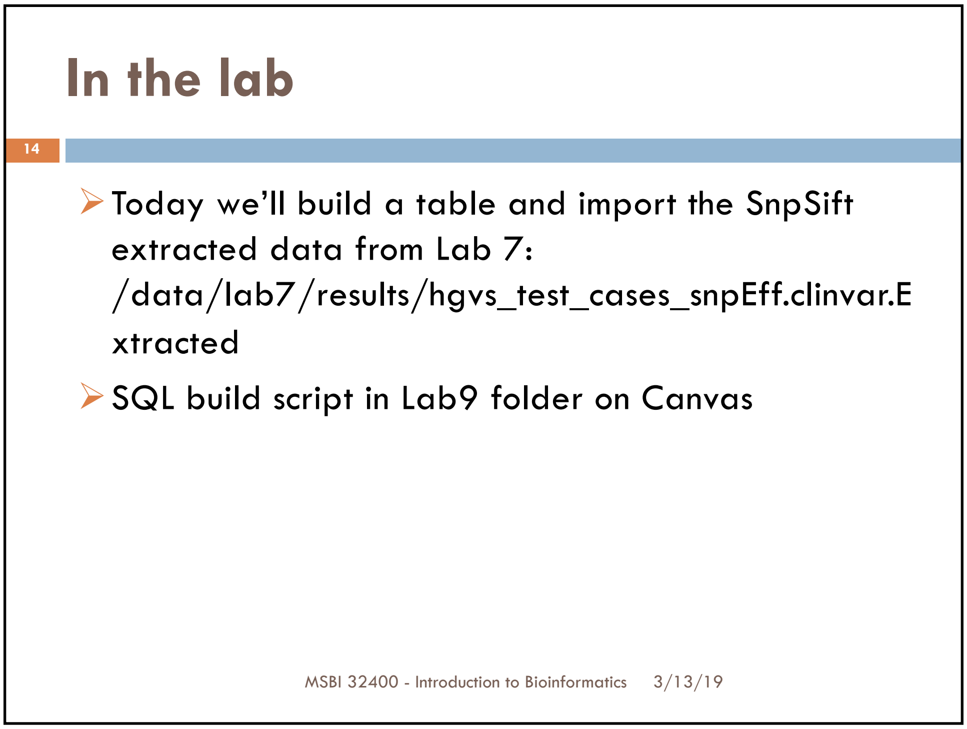
Another Buffalo example

13

Files at /data/bds-files/chapter-13-out-of-memory/code-examples

- From /data/labx/data/code-examples, run **python create_table.py**
  - Run ls -ltr to see what was created
- Run **python load_variants.py variants.txt**
- View: sqlite3 variants.db
  - Note **.header on** and **.mode column** improve readability

MSBI 32400 - Introduction to Bioinformatics 3/13/19

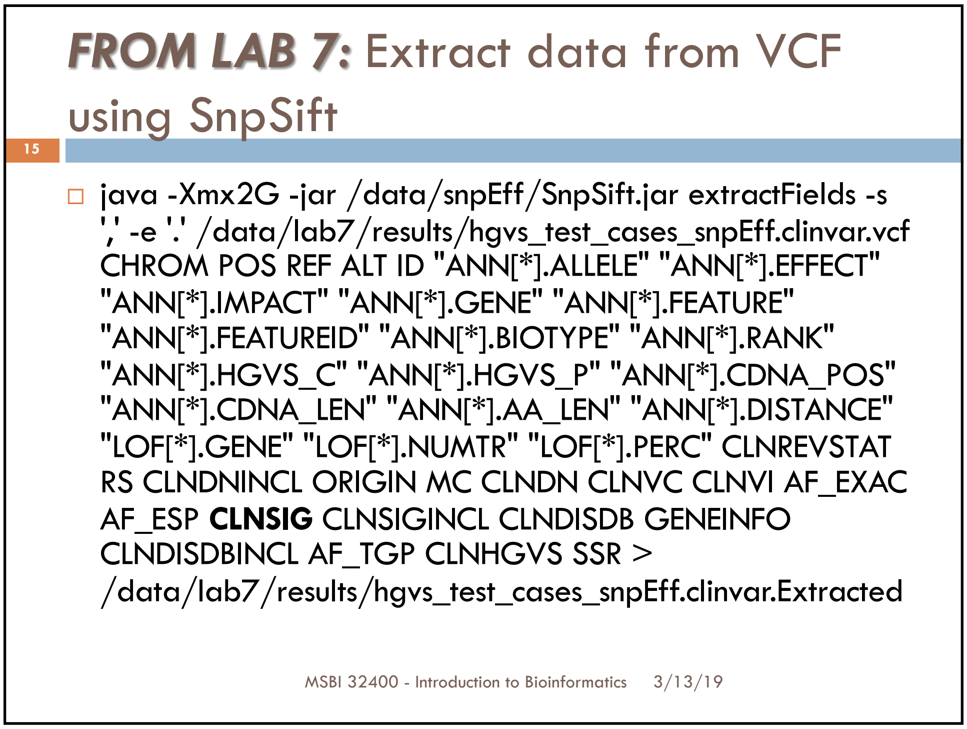


In the lab

14

- Today we'll build a table and import the SnpSift extracted data from Lab 7: /data/lab7/results/hgvs_test_cases_snpEff.clinvar.Extracted
- SQL build script in Lab9 folder on Canvas

MSBI 32400 - Introduction to Bioinformatics 3/13/19

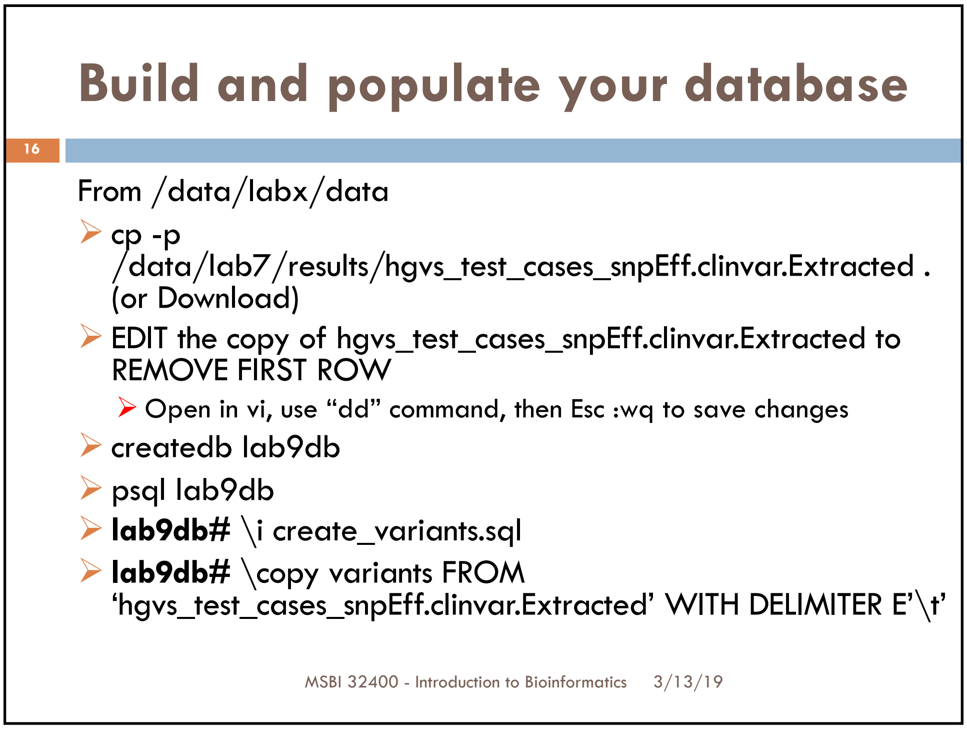


FROM LAB 7: Extract data from VCF using SnpSift

15

```
java -Xmx2G -jar /data/snpEff/SnpSift.jar extractFields -s
',' -e '.' /data/lab7/results/hgvs_test_cases_snpEff.clinvar.vcf
CHROM POS REF ALT ID "ANN[*].ALLELE" "ANN[*].EFFECT"
"ANN[*].IMPACT" "ANN[*].GENE" "ANN[*].FEATURE"
"ANN[*].FEATUREID" "ANN[*].BIOTYPE" "ANN[*].RANK"
"ANN[*].HGVS_C" "ANN[*].HGVS_P" "ANN[*].CDNA_POS"
"ANN[*].CDNA_LEN" "ANN[*].AA_LEN" "ANN[*].DISTANCE"
"LOF[*].GENE" "LOF[*].NUMTR" "LOF[*].PERC" CLNREVSTAT
RS CLNDNINCL ORIGIN MC CLNDN CLNVC CLNVI AF_EXAC
AF_ESP CLNSIG CLNSIGINCL CLNDISDB GENEINFO
CLNDISDBINCL AF_TGP CLNHGVS SSR >
/data/lab7/results/hgvs_test_cases_snpEff.clinvar.Extracted
```

MSBI 32400 - Introduction to Bioinformatics 3/13/19



Build and populate your database

16

From /data/labx/data

- cp -p /data/lab7/results/hgvs_test_cases_snpEff.clinvar.Extracted . (or Download)
- EDIT the copy of hgvs_test_cases_snpEff.clinvar.Extracted to REMOVE FIRST ROW
  - Open in vi, use “dd” command, then Esc :wq to save changes
- createdb lab9db
- psql lab9db
- **lab9db#** \i create_variants.sql
- **lab9db#** \copy variants FROM ‘hgvs_test_cases_snpEff.clinvar.Extracted’ WITH DELIMITER E’\t’

MSBI 32400 - Introduction to Bioinformatics 3/13/19

Now we can ask questions:

17

- SELECT chr, pos, snp_id, rs, af_exac, ann_gene, ann_effect, ann_impact, ann_feature, ann_hgvs_c, ann_hgvs_p FROM variants WHERE clnsig LIKE '%athogenic' AND ann_gene = 'BRCA1';
  - Use wildcard '%' with LIKE instead of =
- SELECT COUNT(*) FROM variants WHERE clnsig LIKE '%athogenic%';

□ Include some example search results in your README

Thanks!

18

□ Please post your lab write-up to Canvas or e-mail Jason with Lab 9 in the subject line. Include answers to the specific lab questions as well as sample SQL statements and a description of the results. E-mail DB questions to me: lhelseth@gmail.com

◆ No homework from next week's lab but we'll still work on it in class since some of you may want to use the bash scripting for your final project