Regular Expressions

The art of pattern matching

m/ATGA{4,7}T(C|A)AAG?TAA/ig

m/L(D|S)Y?(L|C|I)agf+L?DG.g/i

m/^contig-\d{1,}\taugustus_masked\tmatch_part\t(\d{1,})\t(\d{1,})$/

ILLINOIS INSTITUTE OF TECHNOLOGY

Jean-François Pombert, Ph.D.
Office PS 296 (Lab PS 340)

POSIX† *vs.* Perl

SRE/BRE/ERE: Simple/Basic/Extended Regular Expressions

Perl implementation

grep -P 'regxep' *.files

†**P**ortable **O**perating **S**ystem **I**nterface

A few tips…

Be specific but not overly so

Always verify the output

Practice makes perfect

```
## Start simple, then complexify if required
## Regexes may not behave the way you thought
## The difficulty often resides in finding the patterns
```

Online regex testers

Online real-time regular expression testers

http://regexr.com/ ## My favorite

http://regexpal.com/

https://regex101.com/

Copy & paste

Copy the content of Regexp.txt

We’ll use this file for the following slides

Use an online regex tester to practice

Regexp.txt is available in Blackboard

Characters – Regular *vs.* Meta

Regular characters are taken literally

Meta characters have special meanings

The basics

0-9	Numbers
a-z, A-Z	Alphabet (case sensitive)
.	Any character (except newline)

The basics – part II

\d	Digits (numbers)
\w	Words (0-9, a-z, A-Z, _); *i.e.* alphanumeric
\s	Space (any whitespace)

The opposites

Capital letters indicate the opposite of (everything but)

\D	Non-digits
\W	Non-words
\S	Non-whitespace

The quantifiers

*	Zero or more
+	One ore more
?	Zero or one (*i.e.* optional)
{X,Y}	Specific range

Anchors

^	Matches at the start
$	Matches at the end

Boundaries

\bword\b	Full word
\bword	Word starts with
word\b	Word ends with
\Bword\B	Not a boundary word

Other essentials

\t	Tab
\n	Newline

The | alternative

(a|b)	a *or* b	
(a|t|g|c)	a, t, g *or* c	
[atgc]	a, t, g *or* c	## Simpler with multiple characters

Backslashing – the art of escaping

Converts meta to regular characters in searches

\\

\^

\$

The parentheses – Storing values

Assign values for Perl to remember

m/^ATG(\w{3})TAT(\w{3})AAA(GAG)/

$1 $2 $3

TIMTOWTDI

There **I**s **M**ore **T**han **O**ne **W**ay **T**o **D**o **I**t - *Perl's motto*

grep -P 'regex' input(s) > output

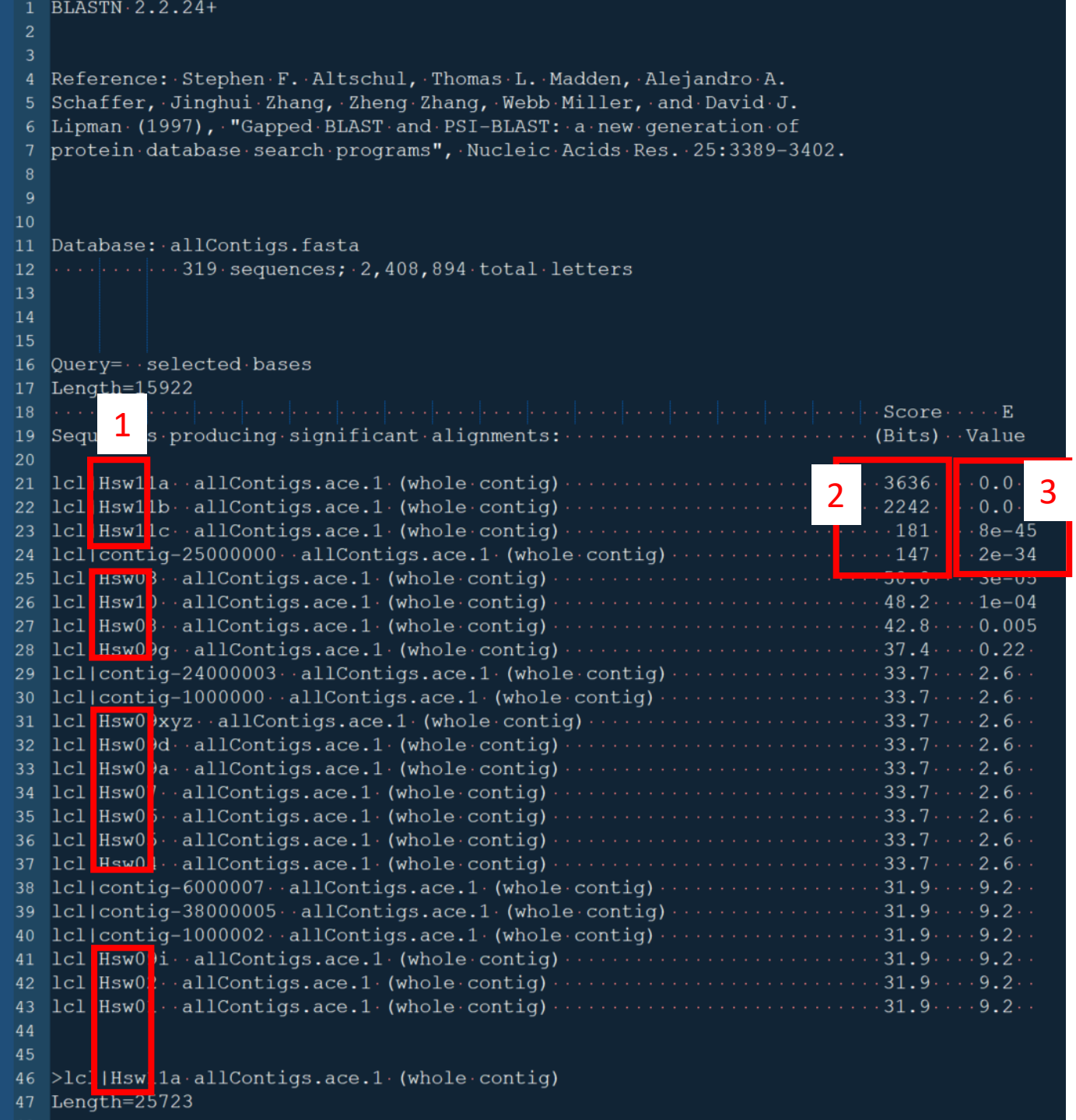
A bit of Perl, without Perl

The -P switch invokes Perl regex mode

diff – compare files line by line

Checking for differences between files

```
diff my_output.txt expected_output.txt        ## 1st file = < ; 2nd file = >
diff -s my_output.txt expected_output.txt     ## -s returns a message if identical
```



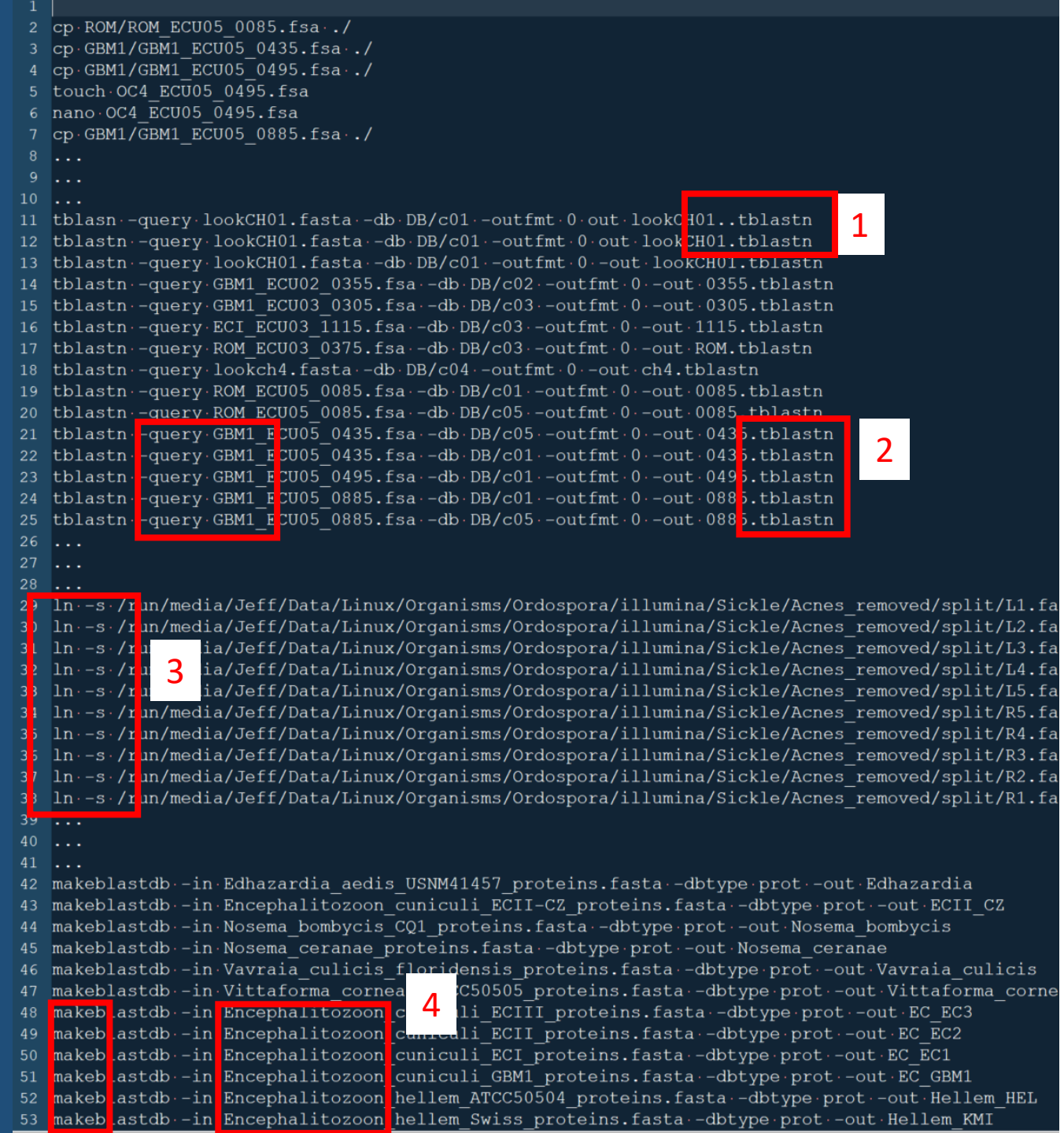
Exercise 1 – From simple to complex

1) Grep all lines containing Hsw
2) Grep lines with score (bits) > 100
3) Grep lines with Evalue < 1e-10

```
## This one can be quite
## complicated! You can skip it and
## do the other exercises first
```

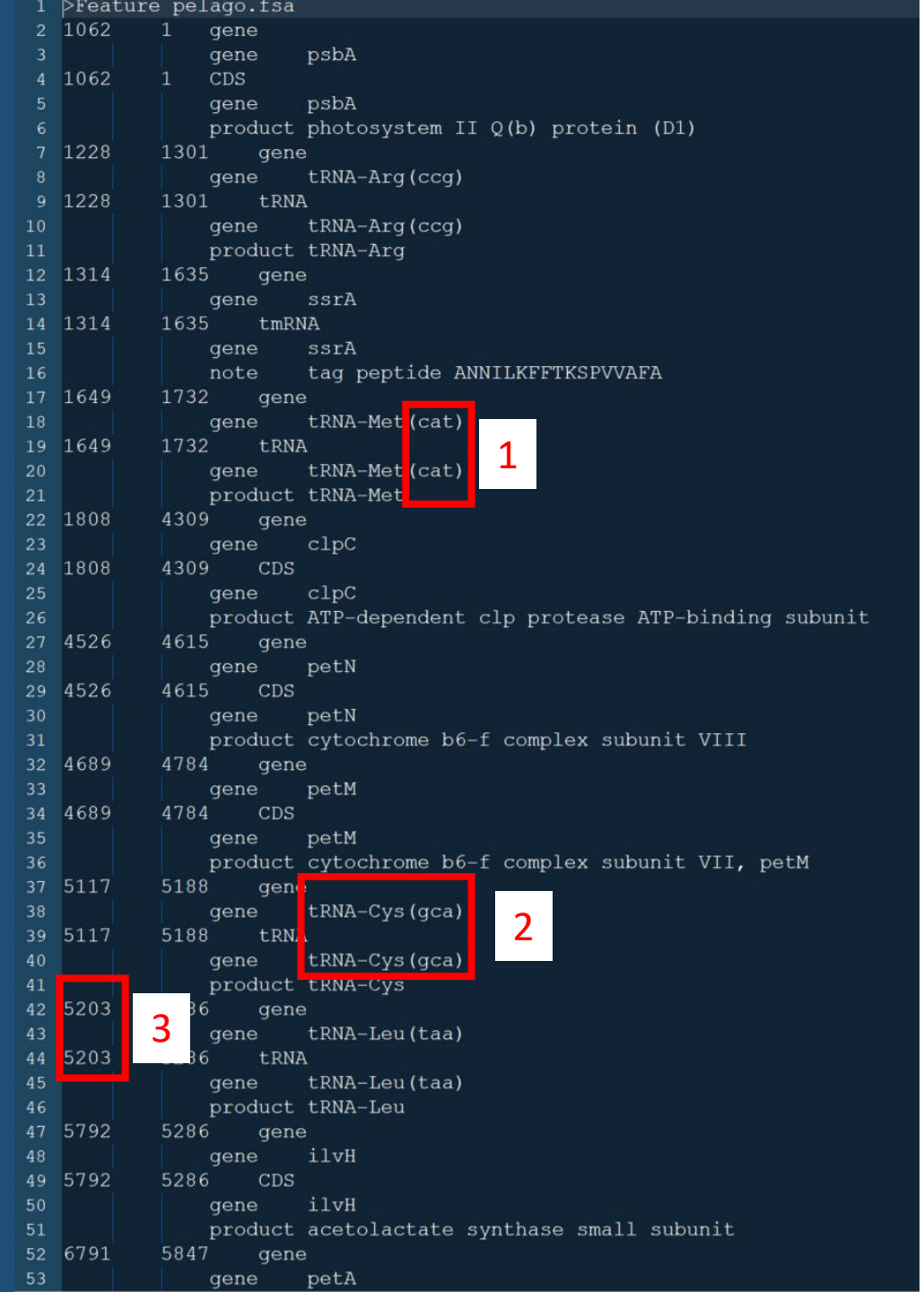
Exercise 2 – Boundaries

1) Grep all words containing ECU
2) Grep only complete words starting with ECU and ending with 0
3) Notice something odd? ## Yep, \w+ does not capture >, only letters, digits and underscores
4) Grep undefined bounded left, product bounded right



Exercise 3 – Start/end lines

1) Grep only lines ending with tblastn
2) Grep only lines ending with tblastn and query starting with GBM1
3) Grep lines starting with ln -s
4) Grep all lines starting with make and containing Encephalitozoon



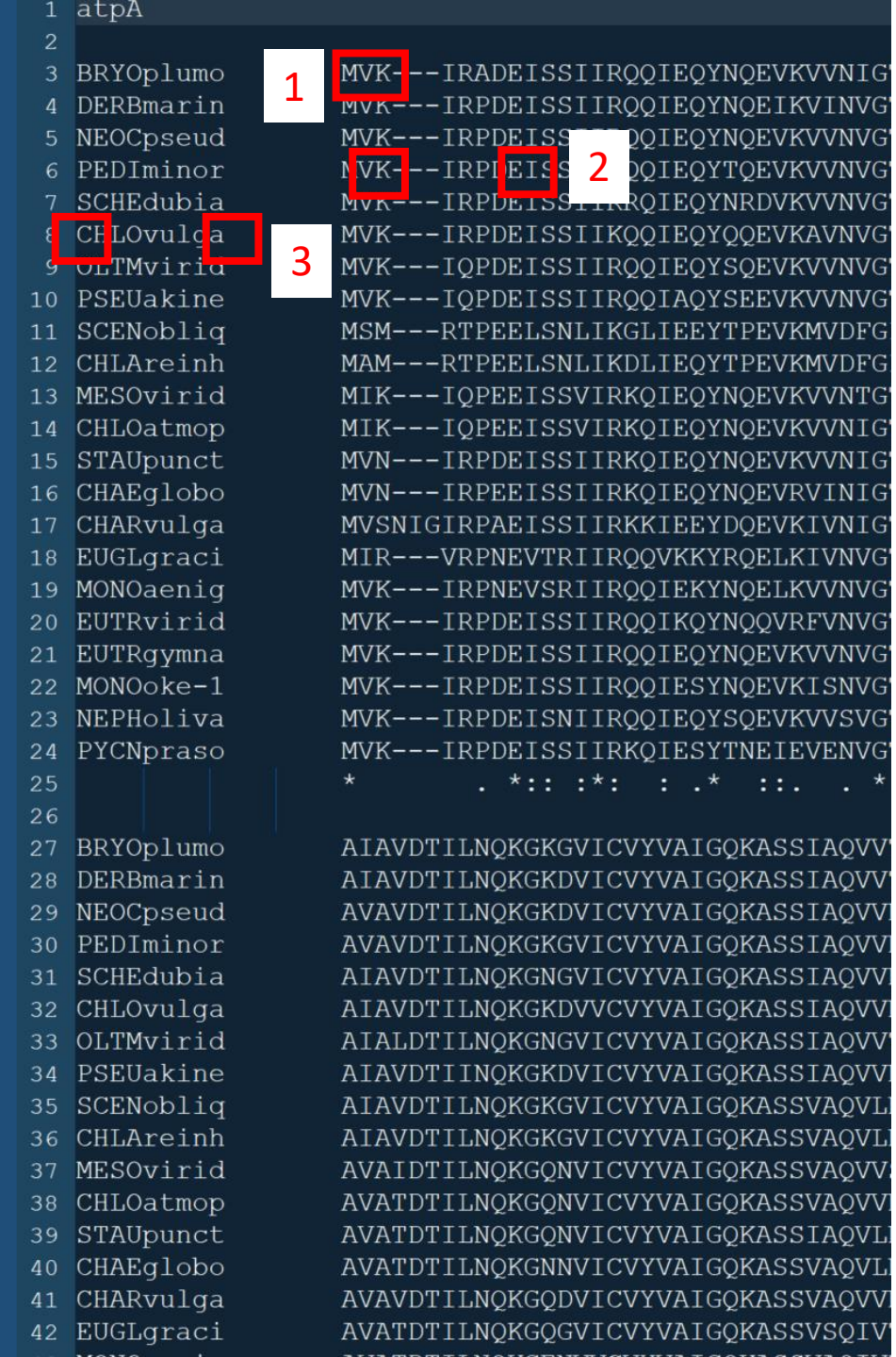
Exercise 4 –
The great escape

1) Grep all lines containing (cat), get the parentheses too

2) Grep all lines containing tRNA-something(anticodon)

3) Can you find the error in those lines? Think biology

4) Grep lines for which with third digit is a zero



Exercise 5 – Another iteration

1) Return sequences with K in 3rd position
2) Return sequences with K or S in 3rd position and I in 9th ## not counting dashes (---)
3) Return names starting with C or O and ending with a or d

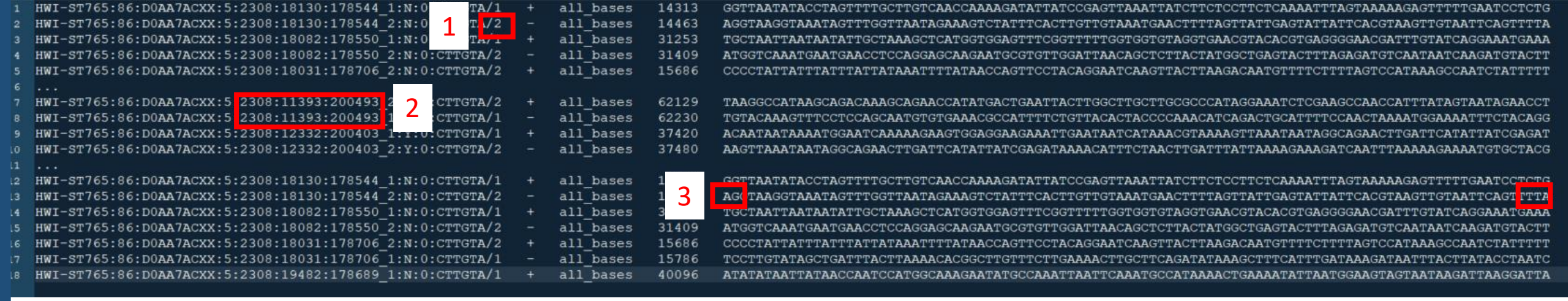
```
XX
RN   [1]
RX   DOI; 10.1093/nar/21.15.3537.
RX   PUBMED; 8346031.
RA   Hallick R.B., Hong L., Drager R.G., Favreau M., Monfort A., Orsat B.,
RA   Spielmann A., Stutz E.;
RT   "Complete sequence of Euglena gracilis chloroplast DNA";
RL   Nucleic Acids Res. 21(15):3537-3544(1993).
XX
...
...
...
FT   rRNA            complement(115617..115732)
FT                   /gene="5S rRNA"
FT                   /product="5S ribosomal RNA"
FT                   /note="rrnC"
FT   rRNA            complement(115817..118693)
FT                   /gene="23S rRNA"
FT                   /product="23S ribosomal RNA"
FT                   /note="rrnC"
FT   tRNA            complement(118718..118790)
FT                   /gene="tRNA-Ala"
FT                   /product="transfer RNA-Ala"
FT                   /anticodon=(pos:118755..118757,aa:Ala)
FT                   /note="tRNA-Ala in rrnC"
FT   tRNA            complement(118800..118873)
FT                   /gene="tRNA-Ile"
FT                   /product="transfer RNA-Ile"
FT                   /anticodon=(pos:118837..118839,aa:Ile)
FT                   /note="tRNA-Ile in rrnC"
FT   CDS             complement(join(2171..3152,3485..3549))
FT                   /transl_table=11
FT                   /gene="ccsA"
FT                   /product="CcsA protein"
FT                   /db_xref="GOA:P31205"
FT                   /db_xref="InterPro:IPR000523"
FT                   /db_xref="InterPro:IPR003593"
FT                   /db_xref="InterPro:IPR011775"
FT                   /db_xref="UniProtKB/Swiss-Prot:P31205"
FT                   /protein_id="CAA50075.1"
FT                   /translation="MNKKTNERPVFPFTSIVGQEEMKLSLILNVIDPKIGGVMIMGDRG
FT                   TGKSTIVRALVDLLPPIDVIENDPYNSDPYDTELMSDDVLEKIKKNEKVSIIQVKTPMV
FT                   VDGLRGDMVTSRAAKALVAFEDRTEVTPKDIFTVITLCLRHRLRKDPLESIDSGYKVQE
FT                   TFKKVFNY"
...
...
...
FT   source          1..143171
FT                   /organism="Euglena gracilis"
FT                   /organelle=plastid:chloroplast
FT                   /strain="Z"
FT                   /mol_type="genomic DNA"
FT                   /db_xref="taxon:3039"
```

1

2

Exercise 6 – EMBL

1) Return CDS, tRNA and rRNA gene features in one command
2) Return author names (hint: look at the file structure)
3) Return strain



Exercise 7 – Sequences

1) Grep all sequence names ending with /2
2) Grep all sequence names containing 2308:11393:200493
3) Grep sequences starting with AG and ending with TA

Self practice

Free time to practice by yourself

Use any text file you want

I'll hang around to answer your questions