

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,})\$/



POSIX[†] *vs.* Perl

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

Perl implementation

`grep -P 'regex' *.files`

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://regexpr.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 Is More Than One Way To Do It - *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
```

```
diff -s my_output.txt expected_output.txt
```

1st file = < ; 2nd file = >

-s returns a message if identical

```

1 BLASTN -2.2.24+
2
3
4 Reference: Stephen F. Altschul, Thomas L. Madden, Alejandro A.
5 Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J.
6 Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of
7 protein database search programs", Nucleic Acids Res. 25:3389-3402.
8
9
10
11 Database: allContigs.fasta
12 ..... 319 sequences; 2,408,894 total letters
13
14
15
16 Query= selected bases
17 Length=15922
18 ..... Score ..... E
19 Sequences producing significant alignments: (Bits) Value
20
21 lcl|Hsw11a|allContigs.ace.1 (whole contig) ..... 3636 ..... 0.0 ..... 1
22 lcl|Hsw11b|allContigs.ace.1 (whole contig) ..... 2242 ..... 0.0 ..... 2
23 lcl|Hsw11c|allContigs.ace.1 (whole contig) ..... 181 ..... 8e-45 ..... 3
24 lcl|contig-25000000|allContigs.ace.1 (whole contig) ..... 147 ..... 2e-34 .....
25 lcl|Hsw03|allContigs.ace.1 (whole contig) ..... 58.8 ..... 5e-05 .....
26 lcl|Hsw10|allContigs.ace.1 (whole contig) ..... 48.2 ..... 1e-04 .....
27 lcl|Hsw03|allContigs.ace.1 (whole contig) ..... 42.8 ..... 0.005 .....
28 lcl|Hsw09g|allContigs.ace.1 (whole contig) ..... 37.4 ..... 0.22 .....
29 lcl|contig-24000003|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
30 lcl|contig-1000000|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
31 lcl|Hsw0xyz|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
32 lcl|Hsw09d|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
33 lcl|Hsw09a|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
34 lcl|Hsw07|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
35 lcl|Hsw06|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
36 lcl|Hsw06|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
37 lcl|Hsw04|allContigs.ace.1 (whole contig) ..... 33.7 ..... 2.6 .....
38 lcl|contig-6000007|allContigs.ace.1 (whole contig) ..... 31.9 ..... 9.2 .....
39 lcl|contig-38000005|allContigs.ace.1 (whole contig) ..... 31.9 ..... 9.2 .....
40 lcl|contig-1000002|allContigs.ace.1 (whole contig) ..... 31.9 ..... 9.2 .....
41 lcl|Hsw09i|allContigs.ace.1 (whole contig) ..... 31.9 ..... 9.2 .....
42 lcl|Hsw02|allContigs.ace.1 (whole contig) ..... 31.9 ..... 9.2 .....
43 lcl|Hsw01|allContigs.ace.1 (whole contig) ..... 31.9 ..... 9.2 .....
44
45
46 >lcl|Hsw11a|allContigs.ace.1 (whole contig)
47 Length=25723

```

Exercise 1 – From simple to complex

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

This one can be quite

complicated! You can skip it and

do the other exercises first

```
1 > ECU01_0010·ECU01_0010·undefined·product·4117:4593·forward·MW:16919
2 MPHTGSQHTLQATPKTAG 1 OKAPEQAIALKKEQQLKNQEGATSHRKAHTEGCHTQKT
3 RMSADKAGLRHRQGSSEV 1 ASTRVEGECSSDGVMMFCMPARGEKASGEARGEDV
4 GSSRESROGTAHKSTCMHTEASL 1 IGKVEDAKT
5 > ECU01_0020·ECU01_0020·undefined·product·4391:5086·reverse·MW:25357
6 MEEDSDILRLCICASVCGELLHC 2 CFPARLLLLCCSFSARLLLLCCSFSARLLLL
7 CCIQLLCPSSSALLFLLCCIRLLCPSSSALLFLLCCIRLLCPSSSALLFLLCCIRLLCPS
8 SSASSAPSALLFLPCRAPSRRLRLLRGELSCPLLPVREPCVCLKSLRLLLFSPPPFV
9 AMPLLCACMCASYVLCPAATLSTSRPPPLPRQRPFLLLPSPACRTSSRRHH
10 > ECU01_0030·ECU01_0030·undefined·product·11158:11529·reverse·MW:13491
11 MDVCTITGMTARGRLRVSRSSRWCVQGGGMYMATRLGCIQREMCEGAVSGLWEEGEDGG
12 TRTQMGKRAREVGLEEGVLLWRTLDLGGVGRKLGSEGQSLSENSEQRSLMRWCGGSS
13 ERR
14 > ECU01_0040·ECU01_0040·undefined·product·11158:13003·forward·MW:22950
15 MCKDRQHTSRPQIQHNRVKTPLAKLTSTIAKGPLHKKSTIMGE 4 YHRCYRRSSKESSA
16 IIPCKTRTYSTVSETAWRQTNPSPNELLSSMLPPVPRRPRGGCRPLHAPLLNKMPQTFPA
17 ASERPMPSSRLSKATQNVQTRPSERPAPCHRRPGRPGGGRDPPEACHPWSLGPGLGLLA
18 PSEVQFDCLEASRTWNTFIGAYTK
19 > ECU01_0050·ECU01_0050·undefined·product·13837:14457·forward·MW:22743
20 MPRPASHLAPMPSDHPDFRSKSARLRCQPPRTNCGTFKQPPSVAATSRPKPGNPFLOPP
21 TKGTTPPKKKKKNHTEGCHTHEANPEPNTKHTETESPKPQTSTQHHTPTITPSSLLSQNT
22 QREKRGLPLLTSRPSTIPANTYQPOSPIHSHSTPLQRPISTALLHQLNHIRARNIRHTGR
23 LHGSPTKGAQTAQQAQPHPPKQLATL
24 > ECU01_0060·ECU01_0060·undefined·product·14529:14834·reverse·MW:10709
25 MRSISVGLDREESGSDGLCLGLVVLATLAAGFVAGSDVLRWGFPGPATGDVCLWVEAGFW
26 RCPPLPGGGADGQGPGEFWDAAALREARMFCRSCVGLRMIGWRR
27 > ECU01_0070·ECU01_0070·undefined·product·15893:16657·forward·MW:28717
28 MNITHVPEIHRDTKQHTENLRHWRKILGIAPFVSIVFPAIMYFISDEDSFKKSLLLRFIT
29 ILLPFSYSAVQYAILHTTPYTLNLLFLAFAAISILSITALPINEWKGDDSLIFIVLP
30 SLFIPPTYLLSTSCRLVPQGTAFTDTGINVLIDILILLCPVSLVLVCKEPEYRLLSAVP
31 FPILILARLLNDRYCPSEKSAPPTAPWRVAILVLILTSALIIYAFMMWTPIAILNGYFGL
32 LHKLRSEFLSLRPD
33 > ECU01_0080·ECU01_0080·undefined·product·17534:18529·reverse·MW:37889
34 MRRYYAAWTLVAAAGVMDCSPRLEKAAAFTLGPDQSQVIVFPFMFGYNIADVLPPTKYGDL
35 KGNARRRVASFLEHNISHAVYFVVGGIAYKDDRSERLFSEMMDGYLKISAGASKVYKG
36 GRKMFSESLTVHEMIFECNKAGDGHVVKYKSIINRLSDMIENALGEVSAEEKRYRRF
37 WSRVKERAGFLYSTERLRRVVEAEKIVCNACKEICLELEEEELMGLLAEGSVRKALKAKV
38 DEDEISRGLYLECTVVNTSLLLDAHREHGGDVTRRELVKQMLLGKKGEEIDRRYINKVANV
39 VKERQRSEMEKRDREQDPERRRLRARRVGSL
40 > ECU01_0090·ECU01_0090·undefined·product·19172:20002·forward·MW:30326
41 MGIIDVQRSHLTATPSKERDAPAHPPPTILPVCILFPYTSIALPVLMYIPEKGQFDQNP
42 FLKLIAILPPCLYSAVQFPLFLGNPESSCTPRPALYATLYLLLDASLLAFSAISILSIA
43 AFTTTEWNSDEVVAVCSTLLPSLLVLPAILLSTSCALTPGSIGFTDSSVDILIDLLMVSL
44 LAAGLTNLNDESWRFFPYICISSLVVLAKLLRKSSSMPPRRDPAPAPAWRIAFAVFLIFGL
45 SMFVYFSILYECCLIFGNHFPWFPSQAPSNDLTNKW
46 > ECU01_0100·ECU01_0100·undefined·product·20249:22108·reverse·MW:72282
47 MGSVHGWIAWGGGLHGS DVEESEGMKKVRKVLKAFSRKLYDSEVERIRTFEKELCIDTR
48 VMIPFIHFGDRVVALPTTRYQDVQDKSEKKYVEGVVMQLRRLVWRLMVWVHVPVGGSSWIES
49 LINEVFATVSRSDPVSILYKARRRSGIRLMDLVMEVFKQNVSMVSEFGQRLARSIEDR
50 MQGIPGSLSPEEKKEEEMWKIKEHGERLCTKERQEEMVRAQKIICDVCAVVEKDEDR
51 MSFIMEVYSRHLCLKIVMPYTDIEVPLISYIDHHKLVTDEKYKSVDIMAEVFKQAFIEH
52 KGIDDESINNAVREVRERKRLEEMREMEERKRREERAKNEEELLRMVEREEREKREKRE
53 KREESKGRGKRGAGAKFEESKFEDGKEFEFGVEAFEEESAEVPIVETAVGARRKKSLKGG
```

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

```

2 cp·ROM/ROM_ECU05_0085.fsa../
3 cp·GBM1/GBM1_ECU05_0435.fsa../
4 cp·GBM1/GBM1_ECU05_0495.fsa../
5 touch·OC4_ECU05_0495.fsa
6 nano·OC4_ECU05_0495.fsa
7 cp·GBM1/GBM1_ECU05_0885.fsa../
8 ...
9 ...
10 ...
11 tblastn -query lookCH01.fasta -db DB/c01 -outfmt 0 -out lookCH01.tblastn
12 tblastn -query lookCH01.fasta -db DB/c01 -outfmt 0 -out lookCH01.tblastn
13 tblastn -query lookCH01.fasta -db DB/c01 -outfmt 0 -out lookCH01.tblastn
14 tblastn -query GBM1_ECU02_0355.fsa -db DB/c02 -outfmt 0 -out 0355.tblastn
15 tblastn -query GBM1_ECU03_0305.fsa -db DB/c03 -outfmt 0 -out 0305.tblastn
16 tblastn -query ECI_ECU03_1115.fsa -db DB/c03 -outfmt 0 -out 1115.tblastn
17 tblastn -query ROM_ECU03_0375.fsa -db DB/c03 -outfmt 0 -out ROM.tblastn
18 tblastn -query lookch4.fasta -db DB/c04 -outfmt 0 -out ch4.tblastn
19 tblastn -query ROM_ECU05_0085.fsa -db DB/c01 -outfmt 0 -out 0085.tblastn
20 tblastn -query ROM_ECU05_0085.fsa -db DB/c05 -outfmt 0 -out 0085.tblastn
21 tblastn -query GBM1_ECU05_0435.fsa -db DB/c05 -outfmt 0 -out 0435.tblastn
22 tblastn -query GBM1_ECU05_0435.fsa -db DB/c01 -outfmt 0 -out 0435.tblastn
23 tblastn -query GBM1_ECU05_0495.fsa -db DB/c01 -outfmt 0 -out 0495.tblastn
24 tblastn -query GBM1_ECU05_0885.fsa -db DB/c01 -outfmt 0 -out 0885.tblastn
25 tblastn -query GBM1_ECU05_0885.fsa -db DB/c05 -outfmt 0 -out 0885.tblastn
26 ...
27 ...
28 ...
29 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/L1.fa
30 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/L2.fa
31 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/L3.fa
32 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/L4.fa
33 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/L5.fa
34 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/R5.fa
35 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/R4.fa
36 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/R3.fa
37 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/R2.fa
38 ln -s /run/media/Jeff/Data/Linux/Organisms/Ordospora/illumina/Sickle/Acnes_removed/split/R1.fa
39 ...
40 ...
41 ...
42 makeblastdb -in Edhazardia_aedis_USNM41457_proteins.fasta -dbtype prot -out Edhazardia
43 makeblastdb -in Encephalitozoon_cuniculi_ECII-CZ_proteins.fasta -dbtype prot -out ECII_CZ
44 makeblastdb -in Nosema_bombycis_CQ1_proteins.fasta -dbtype prot -out Nosema_bombycis
45 makeblastdb -in Nosema_ceranae_proteins.fasta -dbtype prot -out Nosema_ceranae
46 makeblastdb -in Vavraia_culicis_floridensis_proteins.fasta -dbtype prot -out Vavraia_culicis
47 makeblastdb -in Vittaforma_cornea_ATCC50505_proteins.fasta -dbtype prot -out Vittaforma_cornea
48 makeblastdb -in Encephalitozoon_cuniculi_ECIII_proteins.fasta -dbtype prot -out EC_EC3
49 makeblastdb -in Encephalitozoon_cuniculi_ECII_proteins.fasta -dbtype prot -out EC_EC2
50 makeblastdb -in Encephalitozoon_cuniculi_ECI_proteins.fasta -dbtype prot -out EC_EC1
51 makeblastdb -in Encephalitozoon_cuniculi_GBM1_proteins.fasta -dbtype prot -out EC_GBM1
52 makeblastdb -in Encephalitozoon_hellem_ATCC50504_proteins.fasta -dbtype prot -out Hellem_HEL
53 makeblastdb -in Encephalitozoon_hellem_Swiss_proteins.fasta -dbtype prot -out Hellem_KMI

```

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**


```

1 feature pelago.fsa
2 1062 1 gene
3 gene psbA
4 1062 1 CDS
5 gene psbA
6 product photosystem II Q(b) protein (D1)
7 1228 1301 gene
8 gene tRNA-Arg(ccg)
9 1228 1301 tRNA
10 gene tRNA-Arg(ccg)
11 product tRNA-Arg
12 1314 1635 gene
13 gene ssrA
14 1314 1635 tmRNA
15 gene ssrA
16 note tag peptide ANNILKFFTKSPVVAFA
17 1649 1732 gene
18 gene tRNA-Met(cat)
19 1649 1732 tRNA
20 gene tRNA-Met(cat)
21 product tRNA-Met
22 1808 4309 gene
23 gene clpC
24 1808 4309 CDS
25 gene clpC
26 product ATP-dependent clp protease ATP-binding subunit
27 4526 4615 gene
28 gene petN
29 4526 4615 CDS
30 gene petN
31 product cytochrome b6-f complex subunit VIII
32 4689 4784 gene
33 gene petM
34 4689 4784 CDS
35 gene petM
36 product cytochrome b6-f complex subunit VII, petM
37 5117 5188 gene
38 gene tRNA-Cys(gca)
39 5117 5188 tRNA
40 gene tRNA-Cys(gca)
41 product tRNA-Cys
42 5203 5286 gene
43 gene tRNA-Leu(taa)
44 5203 5286 tRNA
45 gene tRNA-Leu(taa)
46 product tRNA-Leu
47 5792 5286 gene
48 gene ilvH
49 5792 5286 CDS
50 gene ilvH
51 product acetolactate synthase small subunit
52 6791 5847 gene
53 gene petA

```

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

```

1 atpA
2
3 BRYOplumo 1 MVK---IRADEISSIIRQQIEQYNQEVKVVNIG
4 DERBmarin MVK---IRPDEISSIIRQQIEQYNQEIKVINVG
5 NEOCpseud MVK---IRPDEISSIIRQQIEQYNQEVKVVNVG
6 PEDImenor MVK---IRPDEISSIIRQQIEQYTQEVKVVNVG
7 SCHEdubia MVK---IRPDEISSIIRQQIEQYNRDVKVVNVG
8 CHLOvulga 3 MVK---IRPDEISSIIRKQIEQYQQEVKAVNVG
9 OLTMvirid MVK---IQPDEISSIIRQQIEQYSQEVKVVNVG
10 PSEUakine MVK---IQPDEISSIIRQQIAQYSEEVKVVNVG
11 SCENobliq MSM---RTPEELSNLIKGLIEEYTPVKMVDVG
12 CHLAreinh MAM---RTPEELSNLIKDLIEQYTPEVKMVDVG
13 MESOvirid MIK---IQPEEISSVIRKQIEQYNQEVKVVNTG
14 CHLOatmop MIK---IQPEEISSVIRKQIEQYNQEVKVVNIG
15 STAUpunct MVN---IRPDEISSIIRKQIEQYNQEVKVVNIG
16 CHAEglobo MVN---IRPEEISSIIRKQIEQYNQEVVRVINIG
17 CHARvulga MVSNIGIRPAEISSIIRKKIEEYDQEVKIVNIG
18 EUGLgraci MIR---VRPNEVTRIIRQQVKKYRQELKIVNVG
19 MONOaenig MVK---IRPNEVSRIIRQQIEKYNQELKVVNVG
20 EUTRvirid MVK---IRPDEISSIIRQQIKQYNQQVRFVNVG
21 EUTRgymna MVK---IRPDEISSIIRQQIEQYNQEVKVVNVG
22 MONOoke-1 MVK---IRPDEISSIIRQQIESYNQEVKISNVG
23 NEPHoliva MVK---IRPDEISNIIRQQIEQYSQEVKVVSVG
24 PYCNpraso MVK---IRPDEISSIIRKQIESYTNEIEVENVG
25 * . *:: :: : .* ::. . *
26
27 BRYOplumo AIAVDITILNQKGKGVICVYVAIGQKASSIAQVV
28 DERBmarin AIAVDITILNQKGKDVICVYVAIGQKASSIAQVV
29 NEOCpseud AVAVDITILNQKGKDVICVYVAIGQKASSIAQVV
30 PEDImenor AVAVDITILNQKGKGVICVYVAIGQKASSIAQVV
31 SCHEdubia AIAVDITILNQKNGVICVYVAIGQKASSIAQVV
32 CHLOvulga AIAVDITILNQKGKDVVCVYVAIGQKASSIAQVV
33 OLTMvirid AIALDITILNQKNGVICVYVAIGQKASSIAQVV
34 PSEUakine AIAVDTIINQKGKDVICVYVAIGQKASSIAQVV
35 SCENobliq AIAVDITILNQKGKGVICVYVAIGQKASSVAQVL
36 CHLAreinh AIAVDITILNQKGKGVICVYVAIGQKASSVAQVL
37 MESOvirid AVAIDTILNQKGQNVICVYVAIGQKASSVAQVV
38 CHLOatmop AVATDITILNQKGQNVICVYVAIGQKASSVAQVV
39 STAUpunct AVATDITILNQKGQNVICVYVAIGQKASSIAQVL
40 CHAEglobo AVATDITILNQKGNNVICVYVAIGQKASSVAQVL
41 CHARvulga AVAVDITILNQKGQDVICVYVAIGQKASSVAQVV
42 EUGLgraci AVATDITILNQKGQGVICVYVAIGQKASSVSQIV

```

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
2 RN      [1]
3 RX      DOI; 10.1093/nar/21.15.3537.
4 RX      PUBMED; 8346031.
5 RA      Halliwick R.B., Hong L., Drager R.G., Favreau M., Monfort A., Orsat B.,
6 RA      Spielmann A., Stutz E.;
7 RT      "Complete sequence of Euglena gracilis chloroplast DNA";
8 RL      Nucleic Acids Res. 21(15):3537-3544(1993).
9 XX
10 ...
11 ...
12 ...
13 FT      rRNA          complement(115617..115732)
14 FT                        /gene="5S rRNA"
15 FT                        /product="5S ribosomal RNA"
16 FT                        /note="rrnC"
17 FT      rRNA          complement(115817..118693)
18 FT                        /gene="23S rRNA"
19 FT                        /product="23S ribosomal RNA"
20 FT                        /note="rrnC"
21 FT      tRNA          complement(118718..118790)
22 FT                        /gene="tRNA-Ala"
23 FT                        /product="transfer RNA-Ala"
24 FT                        /anticodon=(pos:118755..118757,aa:Ala)
25 FT                        /note="tRNA-Ala in rrnC"
26 FT      tRNA          complement(118800..118873)
27 FT                        /gene="tRNA-Ile"
28 FT                        /product="transfer RNA-Ile"
29 FT                        /anticodon=(pos:118837..118839,aa:Ile)
30 FT                        /note="tRNA-Ile in rrnC"
31 FT      CDS           complement(join(2171..3152,3485..3549))
32 FT                        /transl_table=11
33 FT                        /gene="ccsA"
34 FT                        /product="CcsA protein"
35 FT                        /db_xref="GOA:P31205"
36 FT                        /db_xref="InterPro:IPR000523"
37 FT                        /db_xref="InterPro:IPR003593"
38 FT                        /db_xref="InterPro:IPR011775"
39 FT                        /db_xref="UniProtKB/Swiss-Prot:P31205"
40 FT                        /protein_id="CAA50075.1"
41 FT                        /translation="MNKKTNERPVFPFTSIVGQEEMKLSLILNVIDPKIGGVMIMGDRG
42 FT                        TGKSTIVRALVDLPPIDVIENDPYNSDPYDTELMSSDDVLEKIKKNEKVSIIQVKTPMV
43 FT                        VDGLRGDMVTSRAAKALVAFEDRTEVTPKDIFTVITLCLRHRLRKDPLESIDSGYKVQE
44 FT                        TFKKVFNYY"
45 ...
46 ...
47 ...
48 FT      source        1..143171
49 FT                        /organism="Euglena gracilis"
50 FT                        /organelle=plastid:chloroplast
51 FT                        /strain="Z"
52 FT                        /mol_type="genomic DNA"
53 FT                        /db_xref="taxon:3039"

```

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

```

1 HWI-ST765:86:D0AA7ACXX:5:2308:18130:178544_1:N:0:TA/1 + all_bases 14313 GGTAAATATACCTAGTTTTGCTTGTCAACCAAAGATATTATCCGAGTTAAATTATCTTCTCCTTCTCAAAATTTAGTAAAAAGAGTTTTTGAATCCTCTG
2 HWI-ST765:86:D0AA7ACXX:5:2308:18130:178544_2:N:0:TA/2 - all_bases 14463 AGGTAAGGTAAATAGTTTGGTTAATAGAAAGTCTATTTCACTTGTTGTAATGAACTTTATAGTTATTGAGTATTATTCACGTAAGTTGTAATTCAGTTTTTA
3 HWI-ST765:86:D0AA7ACXX:5:2308:18082:178550_1:N:0:TA/1 + all_bases 31253 TGCTAATTAATAATATTGCTAAAGCTCATGGTGGAGTTTCGGTTTTTGGTGGTGTAGGTGAACGTACACGTGAGGGGAACGATTTGTATCAGGAAATGAAA
4 HWI-ST765:86:D0AA7ACXX:5:2308:18082:178550_2:N:0:CTTGTA/2 - all_bases 31409 ATGGTCAAATGAATGAACCTCCAGGAGCAAGATGCGTGTGGATTAACAGCTCTTACTATGGCTGAGTACTTTAGAGATGTCAATAATCAAGATGTACTT
5 HWI-ST765:86:D0AA7ACXX:5:2308:18031:178706_2:N:0:CTTGTA/2 + all_bases 15686 CCCCTATTATTTATTTATTATAAATTTTATAACCAGTTCCTACAGGAATCAAGTTACTTAAGACAATGTTTTCTTTTAGTCCATAAAGCCAATCTATTTTT
6 ...
7 HWI-ST765:86:D0AA7ACXX:5:2308:11393:200493_2:N:0:CTTGTA/2 + all_bases 62129 TAAGGCCATAAGCAGACAAAGCAGAACCATATGACTGAATTACTTGGCTTGCTTGCGCCCATAGGAAATCTCGAAGCCAACCATTATAGTAATAGAACCT
8 HWI-ST765:86:D0AA7ACXX:5:2308:11393:200493_1:N:0:CTTGTA/1 - all_bases 62230 TGTACAAAGTTTCCTCCAGCAATGTGTGAAACGCCATTTCTGTTACACTACCCCAAACATCAGACTGCATTTTCCAACATAAATGGAAAATTTCTACAGG
9 HWI-ST765:86:D0AA7ACXX:5:2308:12332:200403_1:N:0:CTTGTA/1 + all_bases 37420 ACAATAATAAAATGGAATCAAAAAGAAGTGGAGGAAGAAATTGAATAATCATAAACGTAAAAGTTAAATAATAGGCAGAACTTGATTCATATTATCGAGAT
10 HWI-ST765:86:D0AA7ACXX:5:2308:12332:200403_2:N:0:CTTGTA/2 - all_bases 37480 AAGTTAAATAATAGGCAGAACTTGATTCATATTATCGAGATAAAACATTTCTAACTTGATTTATTAAGAAAGATCAATTTAAAAAGAAAATGTGCTACG
11 ...
12 HWI-ST765:86:D0AA7ACXX:5:2308:18130:178544_1:N:0:CTTGTA/1 + all_bases 14313 GGTAAATATACCTAGTTTTGCTTGTCAACCAAAGATATTATCCGAGTTAAATTATCTTCTCCTTCTCAAAATTTAGTAAAAAGAGTTTTTGAATCCTCTG
13 HWI-ST765:86:D0AA7ACXX:5:2308:18130:178544_2:N:0:CTTGTA/2 - all_bases 14463 AGGTAAGGTAAATAGTTTGGTTAATAGAAAGTCTATTTCACTTGTTGTAATGAACTTTATAGTTATTGAGTATTATTCACGTAAGTTGTAATTCAGTTTTTA
14 HWI-ST765:86:D0AA7ACXX:5:2308:18082:178550_1:N:0:CTTGTA/1 + all_bases 31253 TGCTAATTAATAATATTGCTAAAGCTCATGGTGGAGTTTCGGTTTTTGGTGGTGTAGGTGAACGTACACGTGAGGGGAACGATTTGTATCAGGAAATGAAA
15 HWI-ST765:86:D0AA7ACXX:5:2308:18082:178550_2:N:0:CTTGTA/2 - all_bases 31409 ATGGTCAAATGAATGAACCTCCAGGAGCAAGAAATGCGTGTGGATTAACAGCTCTTACTATGGCTGAGTACTTTAGAGATGTCAATAATCAAGATGTACTT
16 HWI-ST765:86:D0AA7ACXX:5:2308:18031:178706_2:N:0:CTTGTA/2 + all_bases 15686 CCCCTATTATTTATTTATTATAAATTTTATAACCAGTTCCTACAGGAATCAAGTTACTTAAGACAATGTTTTCTTTTAGTCCATAAAGCCAATCTATTTTT
17 HWI-ST765:86:D0AA7ACXX:5:2308:18031:178706_1:N:0:CTTGTA/1 - all_bases 15786 TCCTTGTATAGCTGATTTACTTAAACACGGCTTGTTTCTTGAAAACCTTGCTTCAGATATAAAGCTTTTCATTTGATAAAGATAATTTACTTATACCTAATC
18 HWI-ST765:86:D0AA7ACXX:5:2308:19482:178689_1:N:0:CTTGTA/1 + all_bases 40096 AATATAAATTATAACCAATCCATGGCAAAGAATATGCCAAATTAATTCAAATGCCATAAACTGAAAATATTAATGGAAGTAGTAATAAGATTAAGGATTA

```

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