Regular Expressions

A regular expression (aka regex) is a sequence of characters that describe or match a pattern of text. For example, the string **aabb12** could be described as *aabb12*, two a's, *two'bs, then 1, then 2*, or *four letters followed by two numbers*.

Keep in mind that regular expressions differ slightly per language but the theort and application remain the same. We will start with Python.

Metacharacters

Metacharacters are characters that have an alternate meaning rather than a literal meaning. There are many of these and they can be found on the Python regular expression documentation.

For example, we'll take the following characters to search a text file:
– 'A'
– '.'
– '$'

```
import re

text = 'Alan is the coolest man. Ever!$$$'

# compiling a regular expression allows you to reuse the regular expression.
A = re.compile('A')
dot = re.compile('.')
dollar = re.compile('$')

# Any pattern that is matches will be presented as an item within a list.
A.findall(text)
```

```
['A']
```

It matches ‘A’! No surprise here. Now let’s see what the ‘.’ does:

```
dot.findall(text)
```

```
['A',
 'l',
 'a',
 'n',
 ' ',
 'i',
 's',
 ' ',
 't',
 'h',
 'e',
 ' ',
 'c',
 'o',
 'o',
 'l',
 'e',
 's',
 't',
 ' ',
 'm',
 'a',
 'n',
 '.',
 ' ',
 'E',
 'v',
 'e',
 'r',
 '!',
 '$',
 '$',
 '$']
```

It matched everything?! That’s because the ‘.’ is a special character that we will get into later.

Now let’s see what the dollar sign finds us:

```
dollar.findall(text)
```

```
['']
```

Nope, not a bug: '$' did match something but it is also a special character.

Position Metacharacters

This type of regular expression is used to match characters based on where they are located as opposed to what the character means.

Let's take the `$` for example. I told you that it matched something, but nothing was returned in the output. That is because `$` is a special character that matches the end of a line. Let's try an example:

```
text = 'Alan is the coolest man ever, man'

# In regular expressions, backslashes tell the engine to interpret
metacharacters as literals.
pattern = re.compile('man')

pattern.findall(text)
```

```
['man', 'man']
```

This is no surprise, it found both instances of 'man'. Now, let's say we only want to capture 'man' at the end of a string:

```
pattern = re.compile('man$')

pattern.findall(text)
```

```
['man']
```

Trust me, this is grabbing the last man. Don't believe me?

```
text = 'Alan is the coolest man ever'

pattern.findall(text)
```

```
[]
```

Told ya. Now, what happens if we actually do want to match a dollar sign? Well, we can use an escape character, `\` (backslash), before the dollar sign to tell the regex interpreter to use the literal meaning. Let's use the first example:

```
text = 'Alan is the coolest man. Ever!$$$'

pattern = re.compile('$')
pattern.findall(text)
```

```
['']
```

Nothing. Let's try to escape it:

```
pattern = re.compile('\$')
pattern.findall(text)
```

```
['$', '$', '$']
```

Now, what do you think this regular expression do?

```
pattern = re.compile('\$$')
pattern.findall(text)
```

```
['$']
```

Similarly, '^' (caret, but not the kind that rabbits eat) matches the beginning of a line:

```
# IGNORECASE tells the intepreter match the meaning of the character, not
character and case
pattern = re.compile('a', re.IGNORECASE)
```

```
pattern.findall(text)
```

```
['A', 'a', 'a']
```

And now with the caret:

```
pattern = re.compile('^a', re.IGNORECASE)

pattern.findall(text)
```

```
['A']
```

There are also boundary metacharacters. For example, '\b' matches a word that ends with 'ing'. Inversely, '\B' matches a non-boundary word, so it would match the 'ing' in 'things' but not 'thing'. This is useful for specifying substrings or whole words exclusively.

NOTE: some characters are treated as literals even when they are backslached. See here for more deltails:
https://stackoverflow.com/questions/2241600/python-regex-r-prefix
https://stackoverflow.com/questions/21104476/what-does-the-r-in-pythons-re-compiler-pattern-flags-mean

```
text = 'pistol'

# 'is' is surrounded by non-blank characters, not boundaries
pattern = re.compile(r'\bis\b')
print('word boundary',pattern.findall(text))

# However, it is surrounded by non-boundary characters, so it is found by non-
word boundary searches
pattern = re.compile('\Bis\B')
print('non-word boundary',pattern.findall(text))
```

```
word boundary []
non-word boundary ['is']
```

```
text = 'is'

pattern = re.compile(r'\bis\b')
print('word boundary', pattern.findall(text))

pattern = re.compile('\Bis\B')
print('non-word boundary', pattern.findall(text))
```

```
word boundary ['is']
non-word boundary []
```

Single Metacharacters

These metacharacters match specific types of charactes. For example, you can match all alphanumeric characters with '\w' or any whitespace character with '\s'

```
text = '123 abc !@#'

# any number
pattern = re.compile('\d')
pattern.findall(text)
```

```
['1', '2', '3']
```

```
# any alphanumeric character
pattern = re.compile('\w')
pattern.findall(text)
```

```
['1', '2', '3', 'a', 'b', 'c']
```

```
# any non-word character
pattern = re.compile('\W')
pattern.findall(text)
```

```
[' ', ' ', '!', '@', '#']
```

```
# any non-newline character
pattern = re.compile('.')
pattern.findall(text)
```

```
['1', '2', '3', ' ', 'a', 'b', 'c', ' ', '!', '@', '#']
```

Quantifiers

All examples before have been searching for individual characters. Quanitfiers allow you to match repeated patterns.

```
text = 'aa bb cdef 123'

# this looks for every instance of a word character
pattern = re.compile('\w')
pattern.findall(text)
```

```
['a', 'a', 'b', 'b', 'c', 'd', 'e', 'f', '1', '2', '3']
```

```
# the '+' tells the interpreter to look for one or more consecutive characters
pattern = re.compile('\w+')
pattern.findall(text)
```

```
['aa', 'bb', 'cdef', '123']
```

```
# '?' is looking for characters that appear once or not at all. It basically
makes the preceding character optional.
text = 'colour'

pattern = re.compile('colou?r')
print('colour',pattern.findall(text))

text = 'color'
print('color', pattern.findall(text))
```

```
colour ['colour']
color ['color']
```

```
# Asterisk is doesn't require the preceding character to be there, but if it is
it will match it
# will repeat the pattern as many times as is can.
text = 'b'
pattern = re.compile('bo*')
print('b',pattern.findall(text))

text = 'boo'
print('boo',pattern.findall(text))

text = 'boooo!'
print('boooo!', pattern.findall(text))
```

```
b ['b']
boo ['boo']
boooo! ['boooo']
```

```
# curly brackets define a number of times in which the preceding pattern will be
repeated
pattern = re.compile('bo{2}')
text = 'boo'
print('boo',pattern.findall(text))

text = 'boooo!'
print('boooo!', pattern.findall(text))
```

```
boo ['boo']
boooo! ['boo']
```

Character Classes

A character class allows you to match a particular set of user defined characters as oppsed to the predefined metacharacters we went over previously. Think of searching for any vowel.

```
text = 'I like chocolate'

pattern = re.compile('[AEIOUY]', re.IGNORECASE)
print('vowels', pattern.findall(text))
```

```
vowels ['I', 'i', 'e', 'o', 'o', 'a', 'e']
```

Alternatively, **when you use a caret ^ within the square brackets** it will match the **inverse** of those defined in the character class. In this case, it will match all consonants.

```
pattern = re.compile('[^AEIOUY]', re.IGNORECASE)
print('consonants', pattern.findall(text))
```

```
consonants [' ', 'l', 'k', ' ', 'c', 'h', 'c', 'l', 't']
```

You can also specify ranges of characters if they are naturally consecutive by using a hyphen - to separate the beginning and the end:

```
pattern = re.compile('[a-d]', re.IGNORECASE)
print('pattern', pattern.findall(text))
```

```
pattern ['c', 'c', 'a']
```

Alterations

This is essentially just an 'or' statement. An example would be if you are looking for whether a sentence says 'we have ten dollars.' or 'I have ten dollars.' Since the only varying piece of that sentence are the pronouns, you can try to match either.

```
text = 'I have ten dollars'

pattern = re.compile('we|i|they', re.IGNORECASE)
print('I', pattern.findall(text))

text = 'They have ten dollars'
print('They', pattern.findall(text))

text = 'We have ten dollars'
print('We', pattern.findall(text))
```

```
I ['I']
They ['They']
We ['We']
```

Backreferences

Back references/captures allow you to reuse regular expressions and/or patterns that match that regular expression.

Let's give a biological example:
You're looking for a motif that has flanking restriction sites ACTG. The motif can be of any length and any composition of nucleotides but is always flanked by those cuts sites:

```
text = 'ACTGTTTTTTTTTACTG'

# the '\1' is the refence to the first captured pattern.
# All patterns are ennumerated but can also be named.
print('matches:',re.search(r'(ACTG)([GTAC]+)\1', text).groups())

text = 'ATCGCAGCTACGACTGAAAAAAAAAAAAAAACTG'
print('matches:',re.search(r'(ACTG)([ACTG]+)\1', text).groups())
```

```
matches: ('ACTG', 'TTTTTTTTT')
matches: ('ACTG', 'AAAAAAAAAAAAAA')
```

Substitution Mode

There are several ways that one can use regular expressions, though the two main modes are searching and substituting.

Searching/matching will only look for whether or not a pattern is matched. Substituting will replace any pattern that is matched with another pattern. Above we have used only searching, so below will only showcase an example of a substitution.

```
text = 'ACTGTTTTTTTTTACTG'
re.sub(r'ACTG', 'AAAA', text)
```

```
'AAAATTTTTTTTTAAAA'
```

Command Line Examples

Constructing a Regular Expression

Arguably the most difficult task with regular expressions is being able to construct one. Often in cases I am trying extract information that match multiple different patterns or the same pattern but in different lines of text. **In order to create a regular expression, you first need to identify the pattern**. This generally requires some brief understanding of the file format and the relevant information as well as finding a pattern that applies to information that you're interested in.

To give you some insight on how to do this, the examples below are scenarios that I often find myself in that are a perfect fit for regular expressions.

Parsing a GFF file

There are a few tools, such as sed, awk, and grep, that are used for text munging but I happen to use Perl, as it provides a lot of flexibility when doing complex regular expressions and other data munging tasks.

grep can only match patterns (or the inverse of patterns) and cannot be used to replace or transform text. Though limited, it is a nice tool to have for quick searches in files or across filesystems. In this example, all I need is to match a pattern, so grep will suffice.

A GFF file is a standard tab-delimited file format for genomic annotations. Most gff files contain every type of annotated feature for that organism (mRNA, exons, UTRs, motifs, etc.) which are not always relevant to the analysis and may sometimes may actuall interfere with the analysis.

Let's say **I want to extract all annotated mRNAs** from this file. First thing is to understand the gff file format. Reading the documentation will tell you that it is tab delimited and the third column contains the feature type. So looking at the file will sho you:

```
! head -n 20 caenorhabditis_elegans.PRJNA13758.WBPS9.annotations.gff3
```

```
##gff-version 3
##sequence-region I 1 15072434
##sequence-region II 1 15279421
##sequence-region III 1 13783801
##sequence-region IV 1 17493829
```

```
##sequence-region V 1 20924180
##sequence-region X 1 17718942
##sequence-region MtDNA 1 13794
I   BLAT_EST_OTHER  expressed_sequence_match    1   50  12.8    -   .
ID=yk585b5.5.6;Target=yk585b5.5 119 168 +
I   BLAT_Trinity_OTHER  expressed_sequence_match    1   52  20.4    +   .
ID=elegans_PE_SS_GG6116|c0_g1_i1.2;Target=elegans_PE_SS_GG6116|c0_g1_i1 174 225
+
I   inverted    inverted_repeat 1   212 66  .   .   Note=loop 426
I   Genbank assembly_component  1   2679    .   +   .   genbank=FO080985
I   Genomic_canonical   assembly_component  1   2679    .   +   .
Name=cTel33B;Note=Clone:cTel33B,GenBank:FO080985
I   Variation_project_Polymorphism  tandem_duplication  1   11000   .   +   .
variation=WBVar02123961;public_name=WBVar02123961;other_name=cewivar00854884;str
ain=JU533;polymorphism=1;consequence=Coding_exon
I   interpolated_pmap_position  gene    1   559784  .   .   .   ID=gmap:spe-
13;gmap=spe-13;status=uncloned;Note=-21.3602 cM (+/- 1.84 cM)
I   Balanced_by_balancer    biological_region   1   5263413 .   .   .
balancer=Rearrangement:hT3;balancer_type=Translocation;Note=Summary:
Translocation (rigorous proof of reciprocity lacking)%252C moderately well
characterized%252C very stable. Very effective balancer for left portion of LG I
from left end to around let-363%252C and the right portion of LG X from the
right end to between dpy-7 and unc-3. hT3(I)%252C which disjoins from normal LG
I%252C is probably LG X (right) translocated to LG I (right). hT3(X)%252Cwhich
disjoins from normal LG X%252C is probably LG I (left) translocated to LG X
(left).,Growth characteristics: Original isolate marked with dpy-5 and unc-29.
Homozygous inviable%252C probably breaks in let-363 (I). Heterozygotes exhibit
reduced viability%252C low level of X chromosome nondisjunction
(1.2%25).,Handling: Easy to manipulate. Rare exceptional progeny carry one half-
translocation as a free duplication. Recombination frequency in unbalanced
intervals increased on both LG I and LG X.,Recommended use: General
balancing%252C strain maintenance
I   Balanced_by_balancer    biological_region   1   7383197 .   .   .
balancer=Rearrangement:sDp2;balancer_type=Duplication;Note=Summary: Free
duplication%252C well characterized%252C does not recombine with normal
homologues. Very effective balancer for the left portion of LG I from the left
end through unc-15 (just left of unc-13).,Handling: sDp2-bearing males mate and
give some progeny%252C but are slow growing and do not compete well with non-Dp
males in mating.,Recommended use: General balancing%252C strain maintenance%252C
mutant screens.
I   Balanced_by_balancer    biological_region   1   7454088 .   .   .
balancer=Rearrangement:szDp1;balancer_type=Duplication;Note=Summary: Complex
free duplication%252C well characterized%252C does not recombine with normal LG
I. Very effective balancer for the left portion of LG I from the left end
through unc-13. Consists of one half-translocation [szT1(X)] from szT1
maintained in addition to a normal chromosome complement.,Growth
characteristics: Animals carrying two copies of szDp1 apparently inviable.
Duplication strains give rise to spontaneous males through meiotic
nondisjunction of the X chromosome.,Handling: szDp1-bearing males either do not
mate or are infertile.,Recommended use:
I   Balanced_by_balancer    biological_region   1   7454088 .   .   .
balancer=Rearrangement:szT1;balancer_type=Translocation;Note=Summary: Reciprocal
translocation%252C well characterized%252C very stable. Effective balancer for
left portion of LG I through unc-13%252C nearly all of LG X from right end to
```

```
around dpy-3. szT1(I) is large segment of LG X (right) translocated to LG I%252C
disjoins from normal LG I. szT1(X) is LG I (left) translocated to fragment of LG
X (left)%252C disjoins from normal LG X.,Handling: Easy to manipulate. Lon-2
szT1 males mate well. Rare exceptional progeny carry one half-translocation as a
complex free duplication. Gives rise spontaneously to rare apparent lethal
mutations that may represent fusion of szT1(X) and the normal X. Shows threefold
enhanced recombination frequency immediately adjacent to right of LG I
breakpoint and about twofold enhanced frequency in the unc-101 - unc-54
interval.,Recommended use: General balancing%252C strain construction%252C
strain maintenance.
I   Balanced_by_balancer    biological_region   1   8244513 .   .   .
balancer=Rearrangement:hT1;balancer_type=Translocation;Note=Summary: Reciprocal
translocation%252C well characterized%252C very stable. Very effective balancer
for left portion of LG I from the left end through let-80%252C and the left
portion of LG V from the left end through dpy-11. hT1(I) is LG V (left)
translocated to LG I (right)%252C disjoins from normal LG I. hT1(V) is LG I
(left) translocated to LG V (right)%252C disjoins from normal LG V.,Growth
characteristics: Homozygous inviable%252C cause unknown. Arrests at L3. Brood
size in heterozygotes ~75. Easy to manipulate. Heterozygous males mate well.
Rare exceptional progeny carry one half-translocation as a complex free
duplication. Recombination frequency in the unbalanced unc-101 - unc-54 interval
on LG I is increased twofold.,Handling: Easy to manipulate. Heterozygous males
mate well. Rare exceptional progeny carry one half-translocation as a complex
free duplication. Recombination frequency in the unbalanced unc-101 - unc-54
interval on LG I is increased twofold.,Recommended use: General balancing%252C
strain maintenance. mutant screens.
```

Looking at the entire file is daunting and doesn't provide a list of features found within this file. So, we can use a few commands to get a full representation of each feature type described in the file.

```
! cut -f3 caenorhabditis_elegans.PRJNA13758.WBPS9.annotations.gff3 | sort | uniq
```

```
antisense_RNA
assembly_component
base_call_error_correction
binding_site
biological_region
CDS
complex_substitution
conserved_region
deletion
DNAseI_hypersensitive_site
duplication
enhancer
exon
experimental_result_region
expressed_sequence_match
five_prime_UTR
```

```
gene
##gff-version 3
G_quartet
histone_binding_site
insertion_site
intron
inverted_repeat
lincRNA
low_complexity_region
miRNA
miRNA_primary_transcript
mRNA
mRNA_region
nc_primary_transcript
ncRNA
nucleotide_match
operon
PCR_product
piRNA
point_mutation
polyA_signal_sequence
polyA_site
polypeptide_motif
possible_base_call_error
pre_miRNA
promoter
protein_coding_primary_transcript
protein_match
pseudogenic_rRNA
pseudogenic_transcript
pseudogenic_tRNA
reagent
regulatory_region
repeat_region
RNAi_reagent
rRNA
SAGE_tag
scRNA
##sequence-region I 1 15072434
##sequence-region II 1 15279421
##sequence-region III 1 13783801
##sequence-region IV 1 17493829
##sequence-region MtDNA 1 13794
##sequence-region V 1 20924180
##sequence-region X 1 17718942
SL1_acceptor_site
SL2_acceptor_site
snoRNA
SNP
snRNA
substitution
tandem_duplication
tandem_repeat
TF_binding_site
```

```
three_prime_UTR
transcribed_fragment
transcription_end_site
transcript_region
translated_nucleotide_match
transposable_element
transposable_element_insertion_site
tRNA
TSS_region
```

We see that there are a lot of *RNA*s here, and there happen to be two different *mRNA* feature types. So, now that we know the feature type and the format we can construct our regular expression:

```
# The -P tells grep to use perl regular expressions which make things a bit
easier otherwise you'll have
# to escape some characters!

# The pattern we're looking for is mRNA surrounded by tabs/white space
! grep -P '\tmRNA\t' caenorhabditis_elegans.PRJNA13758.WBPS9.annotations.gff3 |
head -n 20
```

```
I	WormBase	mRNA	4116	10230	.	-	.
ID=Transcript:Y74C9A.3;Parent=Gene:WBGene00022277;Name=Y74C9A.3;wormpep=WP:CE281
46;locus=homt-1
I	WormBase	mRNA	11495	16793	.	+	.
ID=Transcript:Y74C9A.2a.1;Parent=Gene:WBGene00022276;Name=Y74C9A.2a.1;wormpep=WP
:CE24660;locus=nlp-40
I	WormBase	mRNA	11495	16837	.	+	.
ID=Transcript:Y74C9A.2a.2;Parent=Gene:WBGene00022276;Name=Y74C9A.2a.2;wormpep=WP
:CE24660;locus=nlp-40
I	WormBase	mRNA	11499	16837	.	+	.
ID=Transcript:Y74C9A.2a.3;Parent=Gene:WBGene00022276;Name=Y74C9A.2a.3;wormpep=WP
:CE24660;locus=nlp-40
I	WormBase	mRNA	11505	16837	.	+	.
ID=Transcript:Y74C9A.2a.4;Parent=Gene:WBGene00022276;Name=Y74C9A.2a.4;wormpep=WP
:CE24660;locus=nlp-40
I	WormBase	mRNA	11618	16837	.	+	.
ID=Transcript:Y74C9A.2a.5;Parent=Gene:WBGene00022276;Name=Y74C9A.2a.5;wormpep=WP
:CE24660;locus=nlp-40
I	WormBase	mRNA	11623	16837	.	+	.
ID=Transcript:Y74C9A.2b;Parent=Gene:WBGene00022276;Name=Y74C9A.2b;wormpep=WP:CE4
9228;locus=nlp-40
I	WormBase	mRNA	17487	26781	.	-	.
ID=Transcript:Y74C9A.4b;Parent=Gene:WBGene00022278;Name=Y74C9A.4b;wormpep=WP:CE2
8147;locus=rcor-1
I	WormBase	mRNA	17497	24796	.	-	.
ID=Transcript:Y74C9A.4d;Parent=Gene:WBGene00022278;Name=Y74C9A.4d;wormpep=WP:CE4
9439;locus=rcor-1
```

```
I	WormBase	mRNA	17497	26643	.	-	.
ID=Transcript:Y74C9A.4c;Parent=Gene:WBGene00022278;Name=Y74C9A.4c;wormpep=WP:CE4
9153;locus=rcor-1
I	WormBase	mRNA	17497	26781	.	-	.
ID=Transcript:Y74C9A.4a;Parent=Gene:WBGene00022278;Name=Y74C9A.4a;wormpep=WP:CE2
4662;locus=rcor-1
I	WormBase	mRNA	27591	32544	.	-	.
ID=Transcript:Y74C9A.5;Parent=Gene:WBGene00022279;Name=Y74C9A.5;wormpep=WP:CE402
91;locus=sesn-1
I	WormBase	mRNA	43733	44677	.	+	.
ID=Transcript:Y74C9A.1;Parent=Gene:WBGene00022275;Name=Y74C9A.1;wormpep=WP:CE344
28
I	WormBase	mRNA	47467	49857	.	+	.
ID=Transcript:Y48G1C.12;Parent=Gene:WBGene00044345;Name=Y48G1C.12;wormpep=WP:CE3
8647
I	WormBase	mRNA	49919	54426	.	+	.
ID=Transcript:Y48G1C.4a;Parent=Gene:WBGene00021677;Name=Y48G1C.4a;wormpep=WP:CE3
0021;locus=pgs-1
I	WormBase	mRNA	52292	54360	.	+	.
ID=Transcript:Y48G1C.4b;Parent=Gene:WBGene00021677;Name=Y48G1C.4b;wormpep=WP:CE4
9150;locus=pgs-1
I	WormBase	mRNA	55293	64066	.	-	.
ID=Transcript:Y48G1C.5;Parent=Gene:WBGene00021678;Name=Y48G1C.5;wormpep=WP:CE394
37
I	WormBase	mRNA	71425	81071	.	+	.
ID=Transcript:Y48G1C.2b.2;Parent=Gene:WBGene00000812;Name=Y48G1C.2b.2;wormpep=WP
:CE49183;locus=csk-1
I	WormBase	mRNA	71425	81063	.	+	.
ID=Transcript:Y48G1C.2b.1;Parent=Gene:WBGene00000812;Name=Y48G1C.2b.1;wormpep=WP
:CE49183;locus=csk-1
I	WormBase	mRNA	71845	80633	.	+	.
ID=Transcript:Y48G1C.2a.1;Parent=Gene:WBGene00000812;Name=Y48G1C.2a.1;wormpep=WP
:CE34405;locus=csk-1
grep: write error
```

Renaming FastQ Files

The filenames for sequences are generally too many characters an too much information, which can be annoying when working with those files. To extract only the relevant information and shorten the filename I often construct a regular expression, coupled with some bash commands, to rename the files how I deem fit.

```
ls /home/at120/badas/2015-03-11_AE52Y-redo
```

```
[0m[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a1.341000000083f5.fastq.gz[0m*
```

```
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a2.3410000000847d.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a3.341000000084f4.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a4.3410000000857c.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a5.341000000085f3.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a6.3410000000867b.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a7.341000000086f2.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
a8.3410000000877a.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
b1.34100000008403.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
b2.3410000000848a.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
b3.34100000008502.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
b4.34100000008589.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
b5.34100000008601.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
b6.34100000008688.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
b7.34100000008700.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
c1.34100000008410.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
c2.34100000008497.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
c3.3410000000851f.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
c4.34100000008596.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
c5.3410000000861e.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
c6.34100000008695.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
c7.3410000000871d.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
d1.3410000000842d.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
d2.341000000084a4.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
d3.3410000000852c.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
d4.341000000085a3.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
d5.3410000000862b.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
d6.341000000086a2.fastq.gz[0m*
```

```
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
d7.3410000000872a.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
e1.3410000000843a.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
e2.341000000084b0.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
e3.34100000008539.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
e4.341000000085bf.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
e5.34100000008638.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
e6.341000000086be.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
e7.34100000008737.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
f1.34100000008447.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
f2.341000000084cd.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
f3.34100000008546.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
f4.341000000085cc.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
f5.34100000008645.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
f6.341000000086cb.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
f7.34100000008744.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
g1.34100000008454.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
g2.341000000084da.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
g3.34100000008553.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
g4.341000000085d9.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
g5.34100000008652.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
g6.341000000086d8.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
g7.34100000008751.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
h1.34100000008460.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
h2.341000000084e7.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
h3.3410000000856f.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
h4.341000000085e6.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
h5.3410000000866e.fastq.gz[0m*
```

```
□[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
h6.341000000086e5.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n01 flu mrt-pcr updated
h7.3410000000876d.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a1.342000000083f2.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a2.3420000000847a.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a3.342000000084f1.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a4.34200000008579.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a5.342000000085f0.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a6.34200000008678.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a7.342000000086ff.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
a8.34200000008777.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
b1.34200000008400.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
b2.34200000008487.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
b3.3420000000850f.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
b4.34200000008586.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
b5.3420000000860e.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
b6.34200000008685.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
b7.3420000000870d.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
c1.3420000000841d.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
c2.34200000008494.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
c3.3420000000851c.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
c4.34200000008593.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
c5.3420000000861b.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
c6.34200000008692.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
c7.3420000000871a.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
d1.3420000000842a.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
d2.342000000084a1.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
d3.34200000008529.fastq.gz□[0m*
```

```
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
d4.342000000085a0.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
d5.34200000008628.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
d6.342000000086af.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
d7.34200000008727.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
e1.34200000008437.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
e2.342000000084bd.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
e3.34200000008536.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
e4.342000000085bc.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
e5.34200000008635.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
e6.342000000086bb.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
e7.34200000008734.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
f1.34200000008444.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
f2.342000000084ca.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
f3.34200000008543.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
f4.342000000085c9.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
f5.34200000008642.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
f6.342000000086c8.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
f7.34200000008741.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
g1.34200000008451.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
g2.342000000084d7.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
g3.34200000008550.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
g4.342000000085d6.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
g5.3420000000865f.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
g6.342000000086d5.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
g7.3420000000875e.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
h1.3420000000846d.fastq.gz□[0m*
□[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
h2.342000000084e4.fastq.gz□[0m*
```

```
[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
h3.3420000000856c.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
h4.342000000085e3.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
h5.3420000000866b.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
h6.342000000086e2.fastq.gz[0m*
[38;5;34m000000000-AE52Y l01n02 flu mrt-pcr updated
h7.3420000000876a.fastq.gz[0m*
```

There is a lot of information here, some of which is intelligible only to Tubo. The only information that I would like to retain are the sample names, consisting of **one lowercase letter followed by one number** and the pairing information, which is **n0 and then a 1 or a 2**.

What I usually do to get started is to work with one filename and construct the pattern from that. Regular expressions are read from left to right, so we should start by grabbing the first bit of information we want, which is the pair information:

```
%%bash
echo "000000000-AE52Y l01n02 flu mrt-pcr updated f3.34200000008543.fastq.gz" |
perl -pe 's/^.+n0([12]).+/$1/g'
```

```
2
```

The syntax for a perl substitution regular expression is as follows:

```
perl -pe '<mode>/<pattern>/<replacement text>/<modifier>'
```

mode can be:
– substitution `s`
– match `m`
– translate `tr`

pattern is your regular expression
replacement text is the text to replace text that matches your regular expression

modifier can be:
– ignore case `i`
– global `g`: will find all patterns, not just the first

and some others. Take a look at the quick reference for perl regular expressions for more information.

Next would be to grab the sample name:

```
%%bash

# It's good to be very specific with regular expressions for multiple reasons.
# 1) It limits the scope of the regular expression
# 2) It makes it easier to read later.
# So I do not NEED to specify 'flu' but it makes for fewer instructions and is
more human readable.

echo "000000000-AE52Y l01n02 flu mrt-pcr updated f3.34200000008543.fastq.gz" \
| perl -pe 's/^.+n0([12])\sflu.+\s([a-z]\d)\..+/$1$2/g'
```

```
2f3
```

Lastly, the sample name and file extension and rearrange the caputred patterns:

```
%%bash
echo "000000000-AE52Y l01n02 flu mrt-pcr updated f3.34200000008543.fastq.gz" \
| perl -pe 's/^.+n0([12])\sflu.+\s([a-z]\d)\..+(fastq.gz)$/$2.r$1.$3/g'
```

```
f3.r2.fastq.gz
```

Now to wrap it up in a for loop and invoke a copy command:

```
%%bash
cd /home/at120/badas/2015-03-11_AE52Y-redo
for i in *gz; do cp "$i" `echo $i | perl -pe 's/^.+n0([12])\sflu.+\s([a-
z]\d)\..+(fastq.gz)$/$2.r$1.$3/g'` ; done

ls
```

```
000000000-AE52Y l01n01 flu mrt-pcr updated a1.341000000083f5.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated a2.3410000000847d.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated a3.341000000084f4.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated a4.3410000000857c.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated a5.341000000085f3.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated a6.3410000000867b.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated a7.341000000086f2.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated a8.3410000000877a.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated b1.34100000008403.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated b2.3410000000848a.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated b3.34100000008502.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated b4.34100000008589.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated b5.34100000008601.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated b6.34100000008688.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated b7.34100000008700.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated c1.34100000008410.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated c2.34100000008497.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated c3.3410000000851f.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated c4.34100000008596.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated c5.3410000000861e.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated c6.34100000008695.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated c7.3410000000871d.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated d1.3410000000842d.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated d2.341000000084a4.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated d3.3410000000852c.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated d4.341000000085a3.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated d5.3410000000862b.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated d6.341000000086a2.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated d7.3410000000872a.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated e1.3410000000843a.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated e2.341000000084b0.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated e3.34100000008539.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated e4.341000000085bf.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated e5.34100000008638.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated e6.341000000086be.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated e7.34100000008737.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated f1.34100000008447.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated f2.341000000084cd.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated f3.34100000008546.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated f4.341000000085cc.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated f5.34100000008645.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated f6.341000000086cb.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated f7.34100000008744.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated g1.34100000008454.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated g2.341000000084da.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated g3.34100000008553.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated g4.341000000085d9.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated g5.34100000008652.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated g6.341000000086d8.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated g7.34100000008751.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated h1.34100000008460.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated h2.341000000084e7.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated h3.3410000000856f.fastq.gz
```

```
000000000-AE52Y l01n01 flu mrt-pcr updated h4.341000000085e6.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated h5.3410000000866e.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated h6.341000000086e5.fastq.gz
000000000-AE52Y l01n01 flu mrt-pcr updated h7.3410000000876d.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a1.342000000083f2.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a2.3420000000847a.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a3.342000000084f1.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a4.34200000008579.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a5.342000000085f0.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a6.34200000008678.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a7.342000000086ff.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated a8.34200000008777.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated b1.34200000008400.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated b2.34200000008487.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated b3.3420000000850f.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated b4.34200000008586.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated b5.3420000000860e.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated b6.34200000008685.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated b7.3420000000870d.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated c1.3420000000841d.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated c2.34200000008494.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated c3.3420000000851c.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated c4.34200000008593.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated c5.3420000000861b.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated c6.34200000008692.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated c7.3420000000871a.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated d1.3420000000842a.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated d2.342000000084a1.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated d3.34200000008529.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated d4.342000000085a0.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated d5.34200000008628.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated d6.342000000086af.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated d7.34200000008727.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated e1.34200000008437.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated e2.342000000084bd.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated e3.34200000008536.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated e4.342000000085bc.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated e5.34200000008635.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated e6.342000000086bb.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated e7.34200000008734.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated f1.34200000008444.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated f2.342000000084ca.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated f3.34200000008543.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated f4.342000000085c9.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated f5.34200000008642.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated f6.342000000086c8.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated f7.34200000008741.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated g1.34200000008451.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated g2.342000000084d7.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated g3.34200000008550.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated g4.342000000085d6.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated g5.3420000000865f.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated g6.342000000086d5.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated g7.3420000000875e.fastq.gz
```

```
000000000-AE52Y l01n02 flu mrt-pcr updated h1.3420000000846d.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated h2.342000000084e4.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated h3.3420000000856c.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated h4.342000000085e3.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated h5.3420000000866b.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated h6.342000000086e2.fastq.gz
000000000-AE52Y l01n02 flu mrt-pcr updated h7.3420000000876a.fastq.gz
a1.r1.fastq.gz
a1.r2.fastq.gz
a2.r1.fastq.gz
a2.r2.fastq.gz
a3.r1.fastq.gz
a3.r2.fastq.gz
a4.r1.fastq.gz
a4.r2.fastq.gz
a5.r1.fastq.gz
a5.r2.fastq.gz
a6.r1.fastq.gz
a6.r2.fastq.gz
a7.r1.fastq.gz
a7.r2.fastq.gz
a8.r1.fastq.gz
a8.r2.fastq.gz
b1.r1.fastq.gz
b1.r2.fastq.gz
b2.r1.fastq.gz
b2.r2.fastq.gz
b3.r1.fastq.gz
b3.r2.fastq.gz
b4.r1.fastq.gz
b4.r2.fastq.gz
b5.r1.fastq.gz
b5.r2.fastq.gz
b6.r1.fastq.gz
b6.r2.fastq.gz
b7.r1.fastq.gz
b7.r2.fastq.gz
c1.r1.fastq.gz
c1.r2.fastq.gz
c2.r1.fastq.gz
c2.r2.fastq.gz
c3.r1.fastq.gz
c3.r2.fastq.gz
c4.r1.fastq.gz
c4.r2.fastq.gz
c5.r1.fastq.gz
c5.r2.fastq.gz
c6.r1.fastq.gz
c6.r2.fastq.gz
c7.r1.fastq.gz
c7.r2.fastq.gz
d1.r1.fastq.gz
d1.r2.fastq.gz
d2.r1.fastq.gz
```

```
d2.r2.fastq.gz
d3.r1.fastq.gz
d3.r2.fastq.gz
d4.r1.fastq.gz
d4.r2.fastq.gz
d5.r1.fastq.gz
d5.r2.fastq.gz
d6.r1.fastq.gz
d6.r2.fastq.gz
d7.r1.fastq.gz
d7.r2.fastq.gz
e1.r1.fastq.gz
e1.r2.fastq.gz
e2.r1.fastq.gz
e2.r2.fastq.gz
e3.r1.fastq.gz
e3.r2.fastq.gz
e4.r1.fastq.gz
e4.r2.fastq.gz
e5.r1.fastq.gz
e5.r2.fastq.gz
e6.r1.fastq.gz
e6.r2.fastq.gz
e7.r1.fastq.gz
e7.r2.fastq.gz
f1.r1.fastq.gz
f1.r2.fastq.gz
f2.r1.fastq.gz
f2.r2.fastq.gz
f3.r1.fastq.gz
f3.r2.fastq.gz
f4.r1.fastq.gz
f4.r2.fastq.gz
f5.r1.fastq.gz
f5.r2.fastq.gz
f6.r1.fastq.gz
f6.r2.fastq.gz
f7.r1.fastq.gz
f7.r2.fastq.gz
g1.r1.fastq.gz
g1.r2.fastq.gz
g2.r1.fastq.gz
g2.r2.fastq.gz
g3.r1.fastq.gz
g3.r2.fastq.gz
g4.r1.fastq.gz
g4.r2.fastq.gz
g5.r1.fastq.gz
g5.r2.fastq.gz
g6.r1.fastq.gz
g6.r2.fastq.gz
g7.r1.fastq.gz
g7.r2.fastq.gz
h1.r1.fastq.gz
```

```
h1.r2.fastq.gz
h2.r1.fastq.gz
h2.r2.fastq.gz
h3.r1.fastq.gz
h3.r2.fastq.gz
h4.r1.fastq.gz
h4.r2.fastq.gz
h5.r1.fastq.gz
h5.r2.fastq.gz
h6.r1.fastq.gz
h6.r2.fastq.gz
h7.r1.fastq.gz
h7.r2.fastq.gz
```

```
! echo "TAAAAAA" | perl -pe 's/([ACTG]){3}//g'
```

```
TAAAAAA
```

```

```

April 6, 2018 Alan Twaddle regex, regular expressions