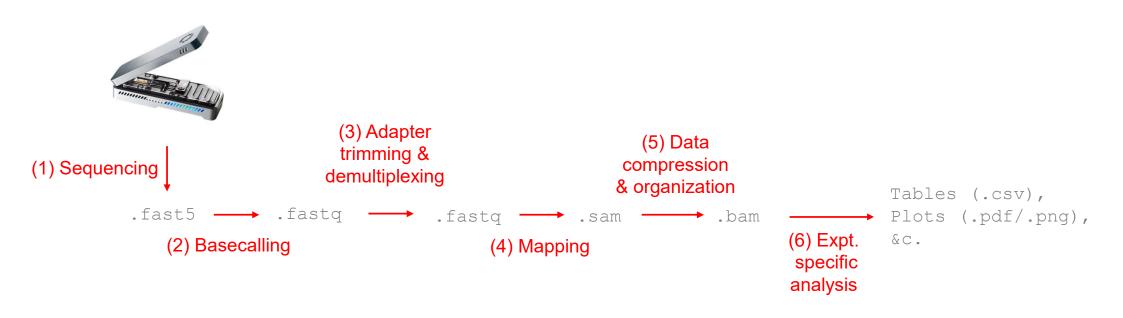# The basics of analyzing
nanopore sequencing data:
extracting quality scores from .fastq files

Jason A. Hendry, 2020/07/25

The data analysis pipeline
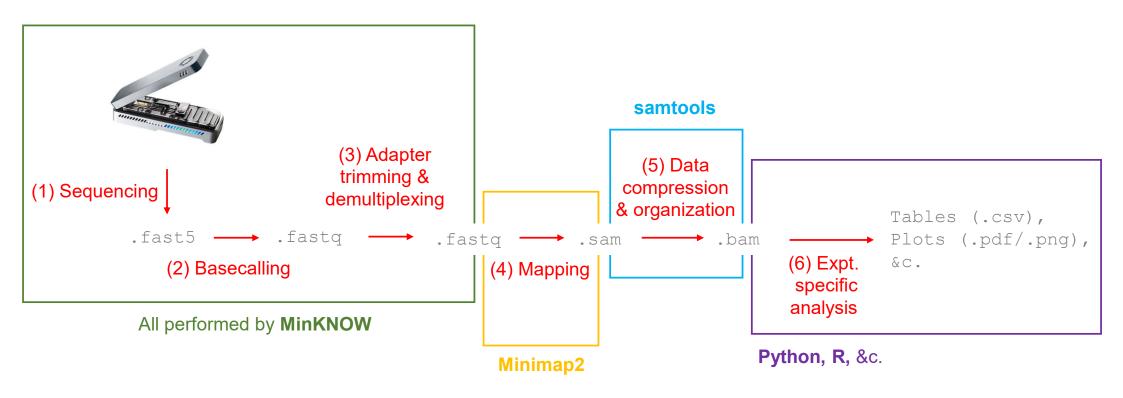## Overview of steps and file types

(3) Adapter
trimming &
demultiplexing

(5) Data
compression
& organization

(1) Sequencing

.fast5 → .fastq → .fastq → .sam → .bam → Tables (.csv),
Plots (.pdf/.png),
&c.

(2) Basecalling

(4) Mapping

(6) Expt.
specific
analysis

The data analysis pipeline

samtools

(3) Adapter trimming & demultiplexing

(1) Sequencing

(5) Data compression & organization

`.fast5` → `.fastq` → `.fastq` → `.sam` → `.bam` → `Tables (.csv), Plots (.pdf/.png), &c.`

(2) Basecalling

(4) Mapping

(6) Expt. specific analysis

All performed by **MinKNOW**

Minimap2

Python, R, &c.

The data analysis pipeline
Overview of steps and file types

Your script(s)! Can be **Python, R, bash,** &c.

samtools

(1) Sequencing

(2) Basecalling

(3) Adapter trimming & demultiplexing

(4) Mapping

(5) Data compression & organization

(6) Expt. specific analysis

.fast5 → .fastq → .fastq → .sam → .bam → Tables (.csv), Plots (.pdf/.png), &c.

All performed by **MinKNOW**
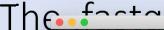
Minimap2

Python, R, &c.

The .fastq
How data on sequencing reads is stored

- Produced directly by the sequencer (Illuminia, nanopore, &c.)

- Just a text file

- Instead of read.txt it is saved as read.fastq

- Stores information about each read:

    - The sequence

    - How confident the sequencer was in calling the sequence

- Stored in standardized way

    - Four lines, repeated for each read

The .fastq

How data on sequencing reads is stored

@read001-2020-09-25-15:33

ATCGTCGCGTTGCTCCGATAGT

+

)('")&+)13;7:<>.&+)-00

@read002-2020-09-25-15:35

TTCGTGGTGAGTGA

+

)('")&+5/)*11*

@read003-2020-09-25-15:37

TTCGTGGTTGTGTTTTTCGTTG

+

)('")&+5/)*11*7;6;31,/

(1) Read Specific Identifier

(2) Nucleotide Sequence

(3) "+"

(4) ASCII-Encoded Error Probabilities

The .fastq

Wha

@e5fd8a58-3a93-4b04-b761-15acb43249bd runid=9ce70c181767ada48688b5f4e975ccd1e61760e7 read=82 ch=293 start_time=2019-04-10T18:29:13Z flow_cell_id=FAK66173 protocol_group_id=NMEC1 sample_id=NMEC1
CTCAATTGTACTTCGTTCAGTTACGTATTGCTAGGGATTAAACAGACGACTACAAACAGAATCGACAGCACCTCTACGATGTGTTTTCTTCCCAAGATAGCATAAATATTTTTATATTTGTCCATATTTCATTATATCAGTAAATGAACTAAAAGGAAATTATACATACATGCATATA
TATATATATATATATATATATATATGGGCACATTTCATTTATTTATTTTTTCTTTCCTAATTAATTCTTACGTTGTTTTCAGAAATGGAAGATTAATATATAGATGATAGACAAAGAAGTGAACAGTGGAGAAAAGGATACCATAGCTATAAATAAAAATAAACAAAATAATTAT
AATGCGTACTAGAATAAAAAAAAAGTGGGAAATATATAATATATAATATTATTATTATATTTTTCTTCTTACATTAATCTTAAATGTCAATCTTATATTTTTTTAAACTATTTCCCTTAATAATGTTTTTATGAAAAACGCTGGCTAAAAAGAAGGAAAAAAAAAAATGATAAA
GACATAGTGTATGTAAAACAGACAGATGTTAATATATAGTAATATTGCATATATATATATACTATATATATATATTGTTGTTATATATTTGTTTTCCTGAATTAAGACTAATTCAGATAAGATTCGTGTTGATTCTTGTGTTCAAAAGATAAATTTCATTCTGCAGGCTGTTGCTATATG
TCTTCTTCAGGATGTTGTATCTAAAAAATTAAATCGTAATAAATAAATAAATGACTATAAATATATATATTTTACTGTTTTTTTTTTGTAGTAAATATTATATACCTATAATATTAAAGCAGAGCATATTAATAGGAATGCACTACAGTGGAAGGACAAATGATTGGATATTTTAGTT
CTTGTAAGACTATTAGAGGCAAATGTTGAACAGATCTAAACATGGATGTAAAAATTTGTAAATTAAACTTCGGTGTCACAGGTTTCTAAAGGTCATGAAATAAATAAATATAAACCGATTGATGTAGCCTGCAGCATACATATATACTATGTCTCATGTAATCTTTATCTGGTGGGTGG
AGGATAGAATTCTCGATTTGGTTCTTTTTAATCAATTTTGCAACAAATGCCAAAATATATAAATAAAAAATATTATCCTTAATCTCTTTAAAAAATAAGTTTAAACACACTTAGGCATTTGCAAGACAGAGCCTGCCCTAAACGTGAGCCATCTATACAAGGGAAAAAAAAAAG
ATTGTATGTGGTTGCCATATATATATCCAATAGGTTGATTTAATCATTGTTTATATGTCATTCCATTTTTGGATGCCCTTCCTTCTTGTGCTAAATTATCTAATGGCAACAATCATTTTGCTTGAATTCTTTTTGATTATTTTTTTACCAGAATTTCATTTGTTATATGTAAG
AAATTAAAAAACAAAATAATAAATAGATATGTGATAAATATGTATTTATTTAATATGGATCTTATAGTCTAAAAATATAAATACTCGGAATGTCGATTTCATCTTGCTGTCATATATGTGGGTAATGAAATAATGCTGGTGCTGTCGATTCCGTTTGTAGTCGTCCTGTTTAACCTTAGCA
ACCATTAACAACTATCC
+
%()0++2,.+23=377;;':3>232;47=4230*'+"/&7>>:2021)++,+13;=3,(+*4112-2(5461.*&"%%%$$'(%0+.+.2/5)/$$#%#"$%$"%&#)(-000.236=6011<422/;5.*<,*+)+(&#%+5)++(*)#181'4+0;<=3=662442291/)++,+--
001.213.,&)()+(&&&#%')1.--,,/))%)%'')2,'&)2/(1950-7>??4-6=64&&.&*%,*:5;4=33-28.548-"%$--()'%&(%(,.$%%%$$#$$&#$$(*/18>2')+#%%%,/'&#',($,34+'20;3<-#"''$+,..5<11>>=200,&%'+75/7>1?11
09201/1%*&$(/13<:?==;???-+"#$$$-*,)+./03184/-/00:55734=-2,-,(*+73=>7/7><63+*,-63=43333<;.038//>3-9)''$$%))+,'/970*&&-2&.$($'%#&$'%.33'("&*567%*"#$%&()*7.0'$*((-13186705;:;8313438.
)#%#&'%%"$"#$'/0+')&#"%$&-+18/4/&'&&&"($(%'*,(*()#%'+*+**))()))()+-,+,*,%,0%/6*262237+53+)/1+'(&$#(#,),-%&%-&-+3$#"%''%(#+(&&'$%##"%$#$%%&$&2-0;<,+1515-<()&).%#"#&)-01&00%$(+*$(%
)+09((/)&*'($&#+$"##%.764;81)*,150)$"&"$%%*)+55<0.0)##$#'$0.47*++-4/1(+'.28*&+/+/7354<:7)$%$#%$)&%&&%#&)*&)''&&&&%()(/21$'&(*(4757876,'/3.;7.$&$#"$'-,./)&,-229:0-(.7/*.,*0)$#""(0
+*1#()+'*''&)##$$&#%*"%%#$.00()'*-)1+*&(+746.(*0/))(0)"&%+-)4-1)&(.,-.0*)*$$")(&+/))'12--/'"')/3913<2./1+)$,.//,'*06/$)&##$"%#'#''$*+)*,--4(%&###$"%"%"#%"%%(",)'''"&#&3*.*+*.
%'+1<9,03:,+"$""#&$$%))+,'$$%)'%(%'%"####&#"%#%&)(,.9980%((.36.-+(-,1161628244327;39??<;3)12<=9?>-+)$%&$$(#&#%%&)"*8./(-+.()&*(+1-)*4--(0'$*6-;:,+"""""%*,0(,-4789:<;<;<==<7/
%,&#%#&#$%%$%*"%%"$#%$('(*.**+&2+,1-82:5/3;+%"###&$*&"%$&#%"$""*,-1119;6+%0+=14,%)0040:.<-&+"*208;.=--.8:>:=30%'#$$+-03814-1<9*'$%$%'88%#()/*332>2244;0(#&$"$&(+45,).82*,.255979+*&
-..61>>><?51:?34-/7677:,#$2+#$"%%''+$$"#"%",62*0/)$+($$'./0$0(()1(,4,-:>8.++(*-,,0.&*/))+//)#)8-,+1/01+'*'+-4923)0*-%('&'*'&(2,+6*,'*&*&.)2390...;>/0..;.'*-2772+/0++:>7775424.&)(
&&'&,$)%%(%&&$""#
@02fbb0ec-2f6a-4776-a625-f187f0c28b4f runid=9ce70c181767ada48688b5f4e975ccd1e61760e7 read=112 ch=237 start_time=2019-04-10T18:29:11Z flow_cell_id=FAK66173 protocol_group_id=NMEC1 sample_id=NMEC1
CTCAGTACTTCGTTCAGTTACGTATTGCTAAGGTTAAACAGACTACAAACGGAATCGACAGCACCTACAACAAATTAAAGATTCCTGTTGATTAAAATATATAGATATATGTGAAGAAATTTTAATATAACAAAGGCGTAAATATTCGTGTTATAATTTCTCCAAGAAAAAAAAAAAA
ATATATATATATATATATATATATTTATTTATTTATTTATTCATTCATTTATTATGTTTTATTAATTAAGTATAATTATGGAAGGAGAAAAAGTAAAACAAAGCAAATAGTATCTCAATTTTTCTATGACGTATGATAGGAATCTGGTGGTAACAGCAATAGTGATGATAAAGCGGAAGT
AGTAGATGATTCTAATTCATTTATGAATTTAACTAGTGATAAAAATGAAAACGGAAAATAATAGTTTCCTTTTAAATAATAGTAGTTATGGAAATGTTAAAGATAGCCTAGAATCCATTGATATGAGTGTATTAGATTCGAACTTTGATAGTAAGATTTTTTACCAAGTAATTTATCAA
GAACATTTAATAATATGTCTAAAGATAATATAGGAAATAAATATTTAAATAAAAAATCACATACAAATGAAAATAATAATAATCATAGCAATAATAATAATCTGACAGCAAATAATATAACTAATAATCTTATTAATAATAATATGAATTCTCCATCAATTATGAA
TACCAACAAATAAGAAGAAATTTTAAATTTAACAAATCTTATAAATGATGATTCTGGATTAAACAATTTAAAAAATTTCAACTGTAAATAATATGAAATGATACTTATGAAAGAAGATTATTGAAACGGAATTAAGAAAGTGATGCTAGTGATTTTTGAAAATATGGTAGGTGATTTCA
AATTACATTTATTAATTGGTTAAAAAAGACACAAATAATTTTATTCGAGAAAAAGATAAATTATTTAAAGATAAGAAAGAACTAGAAATGGAAGAGTACGATTGTACAAAAAATTAGAAAACCGTAAAAATATTGAGAACAGAAATTACATGATGAAAGAAAGAAATTAGATAATTGATA
TATCTAATGGTTATAAACAAATAAAAAGAAAAAAGAGAACATAGGAAACGATTTGATGAAGAAAGATTAAGATTTTACAAGAAATCGATAAAATTAAATTAGTATTATATTTAGAAAAAGAAAATATTATCAAAGAATATAGAGAATGATAAAAAAAAATTGTTGATGCAAATCGCTA
CTGAAACTATGATTGATATTAATGTTGGTGGAGCTATTTTGAAACCTCTAGACATACCTTAACACAACAAAAGATTCATTTTATAGAAGTAGTAAGTGGAAGACATCATGTAACCAGAAGAGATAAACAAGGAAGAATATTCTTAGATAGGGATAGTGAGTTATTTAGAATTATACT
TAACTTCTTAAGAAATCCGTTAATATACCCATAAAAGATTTAGTAAGAAAGTGAAGCCTTGTTGTTGAAGCAGAGATTTTATGGTATTAAATTTTACCTTCCCATTAGTATTTTGTATAGGTGGATTTGATGGTGTAGAATATTTAAATTCGATGAATTATTAGATATTAGTGTCAACAAT
GCTGTGTGTGTAGCACCTATGGTCTACCAAAAAAGCTTATTTTGGAAGTGCTGTATTGAATAATTTCTTATACGTTTTGGTGGTAATAACTATGATTATAAGGCTTTATTTGAAACTTCAAGTGTGATGATCGTTTAGAGATGTATGGTGTGTTTTCAAGTAATTTAAATATACCTAGAAG
AAAAATAATTGTGGTGTTACGTCCAAATGGTAGAATTTATTGTATTGGGGGATATGATGGCTCCTTCTATTATACCGAATGTAGAAGCATATGATCATCGTATGAAAGCATGGGTGAGAGTGGCGCAGATACCCTAGATCATCAGCTATAGTATGAAGAAAAATGAATGGGGTGCTGTTCC
ATATGCCTTATTAGAAGCTAGAAGTTCAGGAGCAGCTTTTAATTACCTTAATCAAATATATGTTGTTGGAGGTATTGATAATGAACATAGCATATTAGATTCCAAGATTGAACAATATCAACCATTTAATAAAAGATGGCAATTTCTAAATGGTGTACCAGAAAAAATGAATTTTGGAG
CTGCCATTGTCAGATTCTTATATAGTACAGGAGGAGAAAATGGCGAAGTTCTAAATTCATGTCATTTCTTTTCACCAGATACAAATGAATGGCAACTTGGCCCATCTTTATTAGTTCCCAGATTTGGTCTCCGTTTAATAGCAAATATATAAAATGAAGAAATTTTAAATGTTCTTATT
TTATATTATAATTTATATAAACAAATAATAAAATATTATTTTTTCTTCTTTGTATTTATTTTCATGCACATTTATAAAAAAAAAAAAAAAAAAAATATATATATATATATATATATATATGTGAGATTATACATACATATTAATAAATATATAAATTTATTTTTAAATACCTTTTATACAC
ATTACTTGATTGTTTATGAATCA
+
&0$##(#+,91./9:10;,),/07:<99:1023197>>=&((((((-/856282>77/-**),,4:.//004687494=0,033;--*0+)&9(411,&///..-29978/0..8.*.1*)/3,/5256>=.;7((063:;>?9;<?:8<97:346=5?>4.<+;/+,.68;9<?>???
423699899740/.-0../02><6+0.*(2*+</),23346;5:>1,1=66/8??.%2/.0-301%1,.3.7-,-<492,0-&489,(%&)5:1,22=155=>118;8<;9;/<4??230:9/)(*.097667/1)'..54=87;<;95:5(-8+)*+)(**)024:+%()1)*&(
)-0)'%)/.2+'&*&+('%/23.57/-()',1/4447/00/???<=976;9)*0)6>>433/-..,=?/9:>>81/$*.10/29;<.?...,3-**+4.85'(+.012>30/.+'-=68733466*(00'/*+3+,/6<7'*0.1=112./.)40().0++,'$&**+*.8=8?100<5;
/1=;9.:5.626..),.1429>>*,-24.11.-&,*-+::?99:?==?<:5:64840?0=3.2>==80/($#%%/+16=47129=409.(/&'+**.35)"$&0*22+2*,2--.36=9899?>883440632127./:444922323-+3*)4001.1333/32801/5+1/,.()-
131:=:;37;=60%)''*3-=?<.%'(*&*$0-9:0/.6./0(<9333*4>-/*/*050=:<<074((6;==7-<9+,846798<;,./*'+%-1-)+0777:9:9.92%(*$$,&(5)%33%%(*'..600$(&*)+.2:;;787:761=>8.1,068-6.8---:=987<?89:
'4(<-20<?600300.1081;.2<:2),-6799)=3*7+>>4.2;4.-%.088/-/6.6.-/(09*/<91503.*0;-*1;==:35=528=;2.+**33),0,)-1,*3:61//91/1==7-,,2/=??91220.'&)/,/)/5,31--///8<;:<.*,3)%)/)(.$(%).07()+'
(,33;3.,2/))17569?.+6?>>8*%*5;=9*$)'+5-1<92.-3,(-&3:-*+-/20*.4*((7+6+.07=<6.0,0"+17,3599-<5---66.5.3&8*-<697:>7*(-6;;+'-69;)+/0-/236+)07:,.)##'*6,.66198<9?>=:/;0,=---.012;,+''*++
46/8>-*-/,207++0-3.;1:;/3+,51...28:>>0'--($&$&)+*,$')487<9<9<.<-+==<,*+8>83<=7(*,4'+$'&%%$//620.-+)(-0===8;:68:730%(+&'*,552:/54,0/2106348?0</-,=:/-<;/:12324:5>54>=6$1'.=/1/2.

The .fastq

@read001-2020-09-25-15:33

ATCGTCGCGTTGCTCCGATAGT

+

)('")&+)13;7:<>.&+)-00

Encodes probability that
nucleotide is **wrong**

Where do they come from?

The .fastq

Nucleotides:

	#1	#2
	A 0.02	A 0.02
	T 0.01	T 0.58
	C 0.96	C 0.01
	G 0.01	G 0.39
	P(error) = 0.04	P(error) = 0.42

The .fastq

@read001-2020-09-25-15:33

ATCGTCGCGTTGCTCCGATAGT

+

)(''')&+)13;7:<>.&+)-00
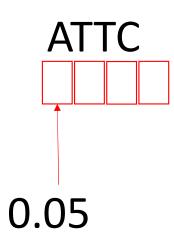
These symbols represent error probabilities

Each nucleotides error probability represented by one character / symbol

How?

Encodes probability that nucleotide is **wrong**
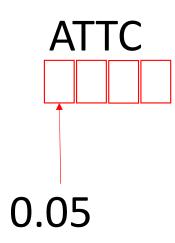
The .fastq

ATTC

□□□□

↑

0.05

The error probability is the probability that the nucleotide is wrong.

Probabilities are between 0 and 1.

They are also continuous e.g. 0.06328782382738232
is a valid probability.

How can we represent them with one character?

The .fastq

ATTC

0.05

The error probability is the probability that the nucleotide is wrong.

Probabilities are between 0 and 1.

They are also continuous e.g. 0.06328782382738232
is a valid probability.

How can we represent them with one character?

Two step solution….

The .fastq

ATTC

0.05

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Q	p
10	0.1
20	0.01
30	0.001
40	0.0001

The .fastq

ATTC

☐☐☐☐

13

0.05

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Q	p
10	0.1
20	0.01
30	0.001
40	0.0001

The .fastq

ATTC

13

0.05

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

ASCII TABLE								
Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100			124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

The .fastq

ATTC

| . | | | |

13

0.05

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

ASCII TABLE								
Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100	|	124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

The .fastq

How can we map probabilities to one character?

ATTC

. ?

?

0.1

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

ASCII TABLE								
Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100			124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

The .fastq

ATTC

. ? □ □

↑

10

↑

0.1

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

ASCII TABLE								
Char	**Dec**	**Binary**	**Char**	**Dec**	**Binary**	**Char**	**Dec**	**Binary**
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100			124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

ATTC

| . | + | | |

10

0.1

Step 1. Convert to quality scores

$$Q = -10\ log_{10}\ p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

ASCII TABLE								
Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100	|	124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

The .fastq
How can we map probabilities to one character?

ATTC

.	+	?	

?

0.01

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

ASCII TABLE

Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100			124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

The .fastq

How can we map probabilities to one character?

ATTC

| . | + | ? | |

Step 1. Convert to quality scores

$$Q = -10 \log_{10} p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

20

0.01

ASCII TABLE

Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100			124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

The .fastq

How can we map probabilities to one character?

ATTC

.	+	5	

20

0.01

Step 1. Convert to quality scores

$$Q = -10 \, log_{10} \, p$$

Step 2. Convert quality scores to ASCII characters

$$C = table(Q + 33)$$

ASCII TABLE								
Char	**Dec**	**Binary**	**Char**	**Dec**	**Binary**	**Char**	**Dec**	**Binary**
!	33	100001	A	65	1000001	a	97	1100001
"	34	100010	B	66	1000010	b	98	1100010
#	35	100011	C	67	1000011	c	99	1100011
$	36	100100	D	68	1000100	d	100	1100100
%	37	100101	E	69	1000101	e	101	1100101
&	38	100110	F	70	1000110	f	102	1100110
'	39	100111	G	71	1000111	g	103	1100111
(	40	101000	H	72	1001000	h	104	1101000
)	41	101001	I	73	1001001	i	105	1101001
*	42	101010	J	74	1001010	j	106	1101010
+	43	101011	K	75	1001011	k	107	1101011
,	44	101100	L	76	1001100	l	108	1101100
-	45	101101	M	77	1001101	m	109	1101101
.	46	101110	N	78	1001110	n	110	1101110
/	47	101111	O	79	1001111	o	111	1101111
0	48	110000	P	80	1010000	p	112	1110000
1	49	110001	Q	81	1010001	q	113	1110001
2	50	110010	R	82	1010010	r	114	1110010
3	51	110011	S	83	1010011	s	115	1110011
4	52	110100	T	84	1010100	t	116	1110100
5	53	110101	U	85	1010101	u	117	1110101
6	54	110110	V	86	1010110	v	118	1110110
7	55	110111	W	87	1010111	w	119	1110111
8	56	111000	X	88	1011000	x	120	1111000
9	57	111001	Y	89	1011001	y	121	1111001
:	58	111010	Z	90	1011010	z	122	1111010
;	59	111011	[	91	1011011	{	123	1111011
<	60	111100	\	92	1011100			124	1111100
=	61	111101	]	93	1011101	}	125	1111101
>	62	111110	^	94	1011110	~	126	1111110
?	63	111111	_	95	1011111	_	127	1111111
@	64	1000000	`	96	1100000			

The .fasta

Wha

@e5fd8a58-3a93-4b04-b761-15acb43249bd runid=9ce70c181767ada48688b5f4e975ccd1e61760e7 read=82 ch=293 start_time=2019-04-10T18:29:13Z flow_cell_id=FAK66173 protocol_group_id=NMEC1 sample_id=NMEC1
CTCAATTGTACTTCGTTCAGTTACGTATTGCTAGGGATTAAACAGACGACTACAAACAGAATCGACAGCACCTCTACGATGTGTTTTCTTCCCAAGATAGCATAAATATTTTTATATTTGTCCATATTTCATTATATCAGTAAATGAACTAAAAGGAAATTATACATACATGCATATA
TATATATATATATATATATATATATGGGCACATTTCATTTATTTATTTTTTCTTTCCTAATTAATTCTTACGTTGTTTTCAGAAATGGAAGATTAATATATAGATGATAGACAAAGAAGTGAACAGTGGAGAAAAGGATACCATAGCTATAAATAAAAATAAACAAAATAATTAT
AATGCGTACTAGAATAAAAAAAAAAGTGGGAAATATATAATATAATATATATTATTATTATATTTTCTTCTTACATTAATCTTAAATGTCAATCTTATATTTTTTAAACTATTTCCCTTAATAATGTTTTTATGAAAAACGCTGGCTAAAAAGAAGGAAAAAAAAAAAATGATAAA
GACATAGTGTATGTAAAACAGACAGATGTTAATATATAGTAATATTGCATATATATATATATATCATATATATATTGTTGTTATATATTTGTTTTCCTGAATTAAGACTAATTCAGATAAGATTCGTGTTGATTCTTGTGTTCAAAAGATAAATTTCATTCTGCAGGCTGTTGCTATATG
TCTTCTTCAGGATGTTGTATCTAAAAAATTAAATCGTAATAAATAAATAAATGACTATAAATATATATATTTTACTGTTTTTTTTTTGTAGTAAATATTATATACCTATAATATTAAAGCAGAGCATATTAATAGGAATGCACTACAGTGGAAGGACAAATGATTGGATATTTTAGTT
CTTGTAAGACTATTAGAGGCAAATGTTGAACAGATCTAAACATGGATGTAAAAATTTGTAAATTAAACTTCGGTGTCACAGGTTTCTAAAGGTCATGAAATAAATAAATATAAACCGATTGATGTAGCCTGCAGCATACATATATACTATGTCTCATGTAATCTTTATCTGGTGGGTGG
AGGATAGATTCTCGATTTGGTTCTTTTTAATACAATTTTGCAACAAATGCCAAAATATAAAATAAAAATATTATCCTTAATCTCTTTAAAAATAAGTTTAAACACACTTAGGCATTTGCAAGACAGAGCCTGCCCTAAACGTGAGCCATCTATACAAGGGAAAAAAAAAAAAAAG
ATTGTATGTGGTTGCCATATATATATATCCAATAGGTTGATTTAATCATTGTTTATATGTCATTCCATTTTTGGATGCCCTTCCTTCTTGTGCTAAATTATCTAATTCTATGGCAACAATCATTTTGCTTGAATTCTTTTTGATTATTTTTTACCAGAATTTCATTTGTTATATGTAAG
AAATTAAAAAACAAAATAATAAAATAGATATGTGATAAATATGTATTTATTTAATATGGATCTTATAGTCTAAAAATATAAATACTCGGAATGTCGATTTCATCTTGCTGTCATATATGTGGGTAATGAAATAATGCTGGTGCTGTCGATTCCGTTTGTAGTCGTCCTGTTTAACCTTAGCA
ACCATTAACAACTATCC
+
%()0++2,.+23=377;;':3>232;47=4230*'+"/&7>>:2021)++,+13;=3,(+*4112-2(5461.*&"%%%$$'(%0+.+.2/5)/$$#%#"$%$"%&#)(-000.236=6011<422/;5.*<,*+)+(&#%+5)++(*)#181'4+0;<=3=662442291/)++,+--
001.213.,&)()+(&&&#%')1.--.,/))%)%')'2),'&)2/(1950-7>??4-6=64&&.&*%,*:5;4=33-28.548-"%$--()'%&(%(,).$%%%$$#$$&#$$(*/18>2')+#%%%,/'&#',($,34+'20;3<-#"''$+,..5<11>>=200,&%'+75/7>1?11
09201/1%*&$(/13<:?==;???-+"#$$$-*,)+././03184//-/00:55734=-2,-,(*+73=>7/7><63+*,-63=43333<;.038//>3-9)''$$%))+,'/970*&&-2&.$($'%#&$'%.33'("&*567%*"#$%&()*7.0'$*((-13186705:;;8313438.
)#%#&'%%"$"#"$'/0+')&#"%$&-+18/4/&'&&&"($(%'*,(*()#%'+*+**))()))()+-,+,*,%,0%/6*262237+53+)/1+'(&$#(#,),-%&%-&-+3$#"%''%(#+(&&'$%##"%$#$%$&$&2-0;<,+1515-<()&).%#"#&)-01&00%$(+*$(%
)+09((/)&*'($&#+$"##%.764;81)*,150)$"&"$%%*)+55<0.0)##$#'$0.47*++-4/1(+'.28*&+/+/7354<:7)$%$#%$)&%&&%#&)*&)''&&&&%()(/21$*'&(*(4757876,'/3.;7.$&$#"$'-,,/)&,-229:0-(.7/*,*0)$#""(0
+*1#()+*''&')##$$&#%"%%#$.00()'*-)1+*&(+746.(*0/))(0)"&%+-)4-1)&(,-,.0*)*$$")(&+/)))'12--/'"')/3913<2./1+)$,,//,'*06/$)&##$"%#'#''$*+)*-,-4(%&##$"%"%'#%"%%(",)''''&#&3*.*+*.
%'+1<9,03:,+"$"""&#$$%))+,'$$%)'%(%'%"####&#"%#%&).(,.9980%((.36.-+(-,1161628244327;39?7<;3)12<=9?>-+)$%&$$(#&#%%&)*8./(-+.()&*(+1-)*4--(0'$*6-;:,+""""%*,0(,-4789:<;<;<==<7/
%,&#%#&#$$%*%"%%"$#%$('(*.*++&2+,1-82:5/3;+%"###&$"&*%&#%"$"#,-1119;6+%0+=14,%)0040:.<-&+"*208;.=-..8:>:=30%'#$$+-03814-1<9*'$%$%'88%#()/*332>2244;0(#&$"$&(+45,).82*,.255979+*&
-..61>><?51:?34-/7677;,#$2+#$"%%''+$$"#"%',62*0/)$+($$'./0$0(()1(,4,-:>8.++(*-,.0.&*/))+//)#)8-,+1/01+'*'+-4923)0*-%('&'*'&(2,+6*,'*&*&.)2390...;>/0..;.'*-2772+/0++:>7775424.&)(
&&'&,$)%%(%&&$""#
@02fbb0ec-2f6a-4776-a625-f187f0c28b4f runid=9ce70c181767ada48688b5f4e975ccd1e61760e7 read=112 ch=237 start_time=2019-04-10T18:29:11Z flow_cell_id=FAK66173 protocol_group_id=NMEC1 sample_id=NMEC1
CTCAGTACTTCGTTCAGTTACGTATTGCTAAGGTTAAACAGACTACAAACGGAATCGACAGCACCTACAACAAATTAAAGATTCCTGTTGATTAAAATATATAGATATATGTGAAGAAATTTTAATATAACAAAGGCGTAAATATTCGTGTTATAATTTCTCCAAGAAAAAAAAAAA
ATATATATATATATATATATATTATTTATTTATTTTATTCATTCATTTATTATGTTTTATTAATTAAGTATAATTATGTGAAGGAGAAAAAGTAAAACAAAGCAAATAGTATCTCAATTTTTCTATGACGTATGATAGGAATCTGGTGGTAACAGCAATAGTGATGATAAAGCGGAAGT
AGTAGATGATTCTAATTCATTTATGAATTTAACTAGTGATAAAAATGAAAACGGAAAATAATAGTTTCCTTTTAAATAATAGTAGTTATGGAAATGTTAAAGATAGCCTAGAATCCATTGATATGAGTGTATTAGATTCGAACTTTGATAGTAAGATTTTTTACCAAGTAATTTATCAA
GAACATTTAATAATATGTCTAAAGATAATATAGGAAATAAATATTTAAATAAAAATCACATACAAATGAAAATAATAATAATCATAGCAATAATAATAATAACTAATAATCTTATTAATAATAATATGAATTCTCCATCAATTATGAA
TACCAACAAATAAGAAGATTTTAAATTTAACAAATCTTATAAATGATGATTCTGGATTAAACAATTTAAAAAATTTCAACTGTAAATAATATGAAATGATACTTATGAAAGAAGATTATTGAAACGGAATTAAGAAAGTGATGCTAGTGATTTTTGAAAATGGTAGGTGATTTA
AATTACATTTATTAATTGGTTAAAAAAGACACAAATAATTTTATTCGAGAAAAAGATAAATTATTTAAAGATAAGAAAGAACTAGAAATGGAAGAGTACGATTGTACAAAAAATTAGAAAACCGTAAAAATATTGAGAACAGAAATTACATGATGAAAGAAAGAAATTAGAGATATTGATA
TATCTAATGGTTATAAACAAATAAAAAAGAAAAAAGAGAACATAGGAAACGATTTGATGAAGAAAGATTAAGATTTTACAAGAAATCGATAAAATTAAATTAGTATTATATTTAGAAAAAGAAAATATTATCAAAGAATATAGAGAATGATAAAAAAAAATTGTTGATGCAAATCGCTA
CTGAAACTATGATTGATATTAATGTTGGTGGAGCTATTTTGAAACCTCTAGACATACCTTAACACAACAAAAGATTCATTTTATAGAAGTAGTAAGTGGAAGACATCATGTAACCAGAAGAGATAAACAAGGAAGAATATTCTTAGATAGGGATAGTGAGTTATTTAGAAATTATACT
TAACTTCTTAAGAAATCCGTTAATATACCCATAAAAGATTTAGTAAGAAAGTGAAGCCTTGTTGTTGAAGCAGAGATTTTATGGTATTAAATTTTACCTTCCCATTAGTATTTTGTATAGGTGGATTTGATGGTGTAGAAATTTAAATTCGATGAATTATTAGATATTAGTCAACAAT
GCTGTGTGTGTACACCTATGGTCTACCAAAAAGCTTATTTTGGAAGTGCTGTATTGAATAATTTCTTATACGTTTTGGTGGTAATAACTATGATTATAAGGCTTTATTTGAAACTTCAAGTGTGATGATCGTTTAGAGATGTGATGGTGTTTTCAAGTAATTTAAATATACCTAGAAG
AAAATAATTGTGGTGTTACGTCCAAATGGTAGAATTTATTGTATTGGGGGATATGATGGCTCTTCTTATTATACCGAATGTAGAAGCATATGATCATCGTATGAAAGCATGGGTGAGAGTGGCGCAGATACCCTAGATCATCAGCTATAGTATGAAGAAAATGAATGGGGTGCTGTTCC
ATATGCCTTATTAGAAGCTAGAAGTTCAGGAGCAGCTTTTAATTACCTTAATCAAATATATGTTGTTGGAGGTATTGATAATGAACATAGCATATTAGATTCCAAGATTGAACAATATCAACCATTTAATAAAAGATGGCAATTTCTAAATGGTGTACCAGAAAAAATGAATTTTGGAG
CTGCCATTGTCAGATTCTTATATAGTACAGGAGGAGAAAATGGCGAAGTTCTAAATTCATGTCATTTCTTTTCACCAGATACAAATGAATGGCAACTTGGCCCATCTTTATTAGTTCCCAGATTTGGTCTCCGTTTAATAGCAAATATATAAAATGAAGAAATTTTAAATGTTCTTATT
TTATATTATAATTTATATAAACAAAATATAAAATATTATTTTTCTTCTTTGTATTTATTTTCATGCACATTTATAAAAAAAAAAAAAAAAAAAAATATATATATATATATATATATATGTAGATTATACATACATATTAATAAATATATAATTTATTTTTAAATACCTTTTATACAC
ATTACTTGATTGTTTATGAATCA
+
&0$##(#+,91./9:10;,),/07:<99:1023197>>=&((((((-/856282>77/-**),,4:.//004687494=0,033;--*0+)&9(411,&///,.-29978/0..8.*.1*)/3,/5256>=.;7((063:;>?9;<?:8<97:346=5?>4.<+;/+,.68;9<>?>??
423699898974d/.-0../02><6+0.*(2*+</),23346;5:>1,1=66/8??.%2/.0-301%1,.3.7-,-<492,0-&489,(%&)5:1,22=155=>118;8<;9;/<4??230:9/)(*.097667/1)'..54=87;<;95:5(-8+)*+)(**)024:+%()1)*&(
)-0)'%)/.2+'&*&+('%/23.57/-()',1/4447/00/???<=976;9)*0)6>>433/-.,=?/9:>>81/$*.10/29;<.?...,3-**+4.85'(+.012>30/.+'-=68733466*(00'/*+3+,/6<7'*0.1=112./.)40().0++,'$&***.8=8?100<5;
/1=;9.:5.626..).,1429>>*,-24.11,-&,*-+::?99:?==?<:5:64840?0=3.2>==80/($#%%/+16=47129=409.(/&'+**.35)"$&0*22+*2*,2--.36=9899?>883440632127./:444922323-+3*)4001.1333/32801/5+1/,.()-
131:=:;37;=60%)''*3-=?<.%'(*&*$0-9:0/.6./0(<9333*/4>-/*/*050=:<<074((6;==7-<9+,846798<:,./*'+%-1-)+0777:9:9.92%(*$$,&(5)%33%%(*'..600$(&*)+.2:::787:761=>8.1,068-6.8---:=987<?89:
'4(<-20<?600300.1081;.2<:2),-6799)=3*7+>>4.2;4.-%.088/-/6.6.-/(09*/<91503.*0;-*1;==:35=528=;2.+**33),0,)-1,*3:61//91/1==7-,,2/=??91220.'&)/,/)/5,31-///8<;:<.*,3)%)/)(.$(%).07()+
(,33;3.,2/))17569?.+6?>>8*%*5;=9*$)'+5-1<92.-3,(-&3:-*+-/20*.4*((7+6+.07=<6.0,0"+17,3599-<5---66.5.3&8*-<697:>7*(-6;;+'-69;)+/0-/236+)07:,,)##'*6,.66198<9?>=:/;0,=--.012;,+''*++
46/8>-*-/,207++0-3,;1::/3+,51...28:>>0'--($&$&)+*,$')487<9<9<.<-+==<,*+8>83<=7(*,4'+$'&%%&$//620.-+)(-0===8:;68:730%(+&'*,552:/54,0/2106348?0</-,=:/-<;/:12324:5>54>=6$1'.=/1/2.

The .fastq
Summary

- Text file produce by the sequencer

- Contains information on reads

- Four lines per read

    - ID

    - Sequence

    - "+"

    - Error probabilities, encoded in ASCII characters