TCR Analysis – Principles, Workflows, and Software

Ben K. Margetts

UCL ICH
& Great Ormond Street Hospital

TCR Analysis Handover – 26th March

AutoTCR

AutoTCR is our lab's **automated sample processing** and **analysis** pipeline

The **data structure** it produces forms **the basis** for all of our **figures and analyses**

To date, our lab has generated **> 600 GB** of compressed TCR sequencing data
That is equal to ~ **600,000 word documents** or **15,000,000 text files**

TCR & Data Science/Biology

Without the 'data science' angle of this work, we would have **no results**.
Continued development of this is **essential** to being a functional TCR lab.

Lab Biologist – Generate the data
Data Biologist – Make the data meaningful

TCR & Data Science/Biology

Statistics + Computer Science → Data Science

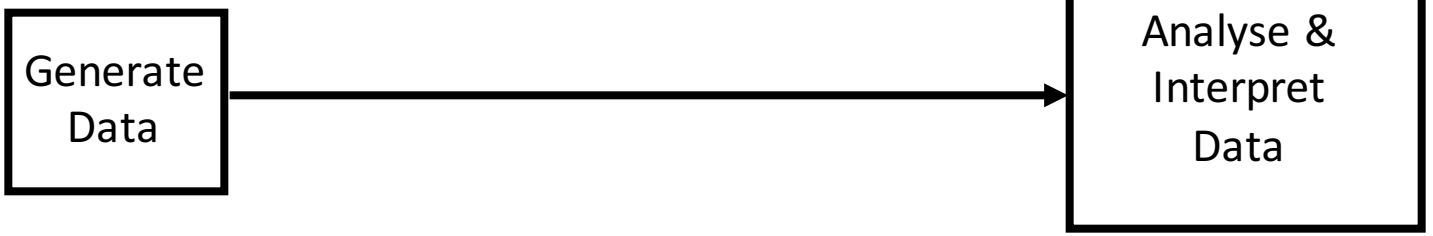
Without the 'data science' angle of this work, we would have **no results**. Continued development of this is **essential** to being a functional TCR lab.

Lab Biologist – Generate the data

Lab Biologist

The way we used to do science

Lab Biologist



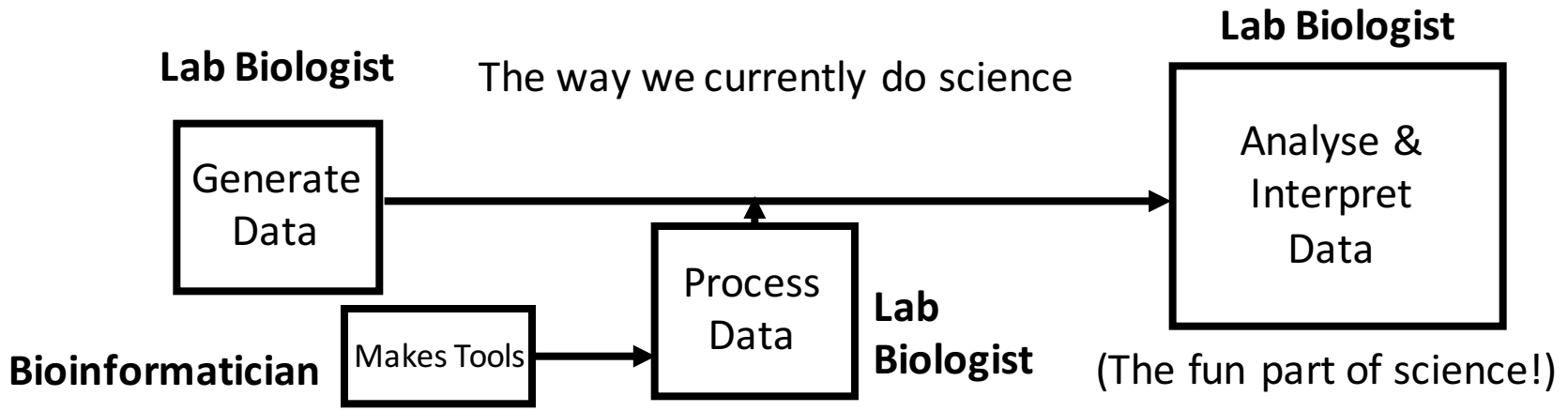
(The fun part of science!)

TCR & Data Science/Biology

Statistics + Computer Science → Data Science

Without the 'data science' angle of this work, we would have **no results**.
Continued development of this is **essential** to being a functional TCR lab.

Lab Biologist – Generate the data
Bioinformatician – Develops methods and tools for Biologists



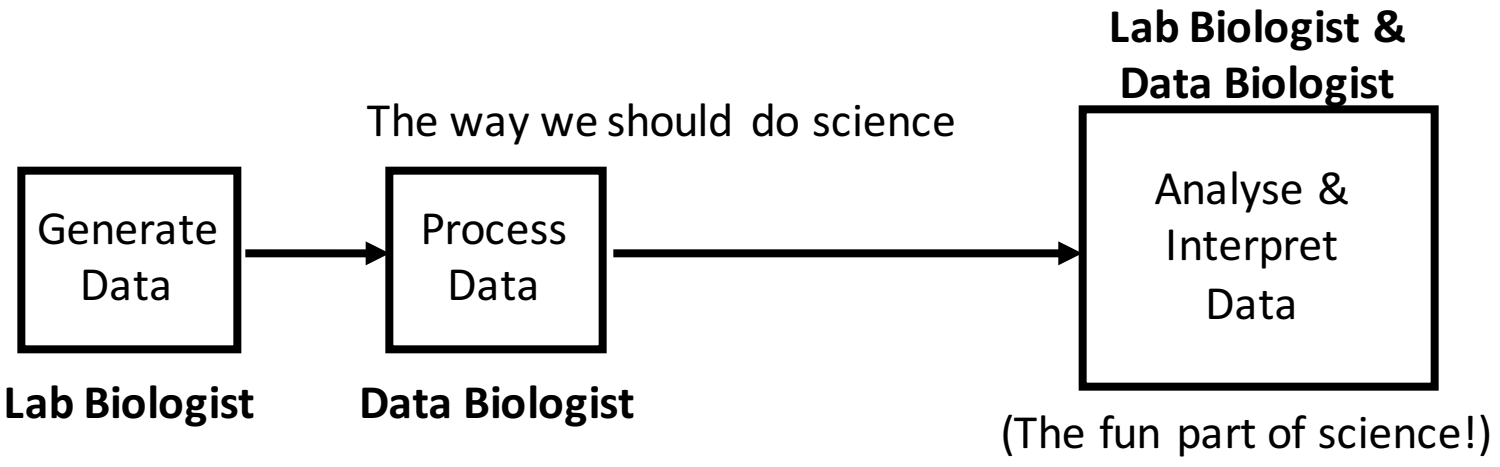
TCR & Data Science/Biology

Statistics + Computer Science → Data Science

Without the 'data science' angle of this work, we would have **no results**. Continued development of this is **essential** to being a functional TCR lab.

Lab Biologist – Generate the data
Data Biologist – Make the data meaningful

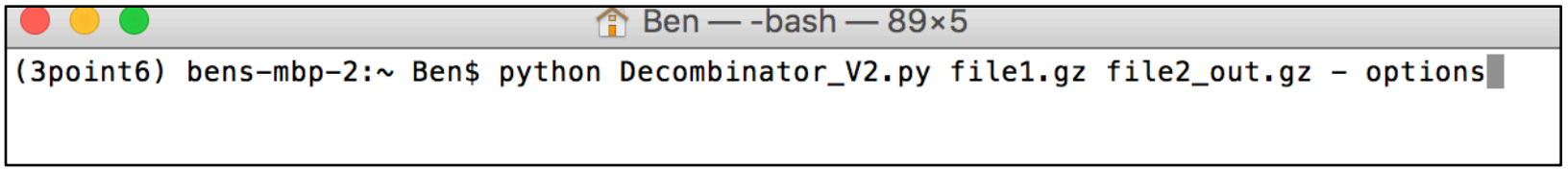
The way we should do science



Why Make The Script?

Before (est. time = days):

- Manually run each file through each of the 5 Decombinator Scripts (12 files per run)



- Validate that no mistakes have been made
- Read in data to analysis platform (Excel/R/Python/SAS/STATA)
- Assign metadata to each file (time, patient, disease, etc)
- Structure data into a dataframe
- Produce data visualisations

Now (est. time = 3 ½ hours):

- Provide correct name in index file.
- Leave AutoTCR running for ~ 3 hours.
- Done.

Unfortunately,

Hadley Wickham – *'You can't do data science in a GUI'*

Handover

	Python	R	R Studio & Tidyverse	Bash
1. Installation & Requirements				
2. Automation and File Structure				
3. Reproducibility				
4. Decombinator a) Demultiplexor b) Decombinator c) CDR3Translator d) GNTranslator				
5. Building a Dataframe				
6. Automated Analysis				
7. Debugging/Common Issues				
8. Subsampling				

Installation & Requirements

- Python Version 2.7
(Anaconda distribution https://www.anaconda.com/download/#macos)

- Python packages (using Conda package manager)
(see https://github.com/innate2adaptive/Decombinator#installation)

- R
(see https://cloud.r-project.org)

- R Studio
(see https://www.rstudio.com/products/rstudio/download/)

- R packages (installed by AutoTCR script)

- Mac OSX only - command line tools
(type *xcode-select –install* in Terminal)

- If in doubt, please ask.

Automation & File Structure

What do you need to do to run AutoTCR?

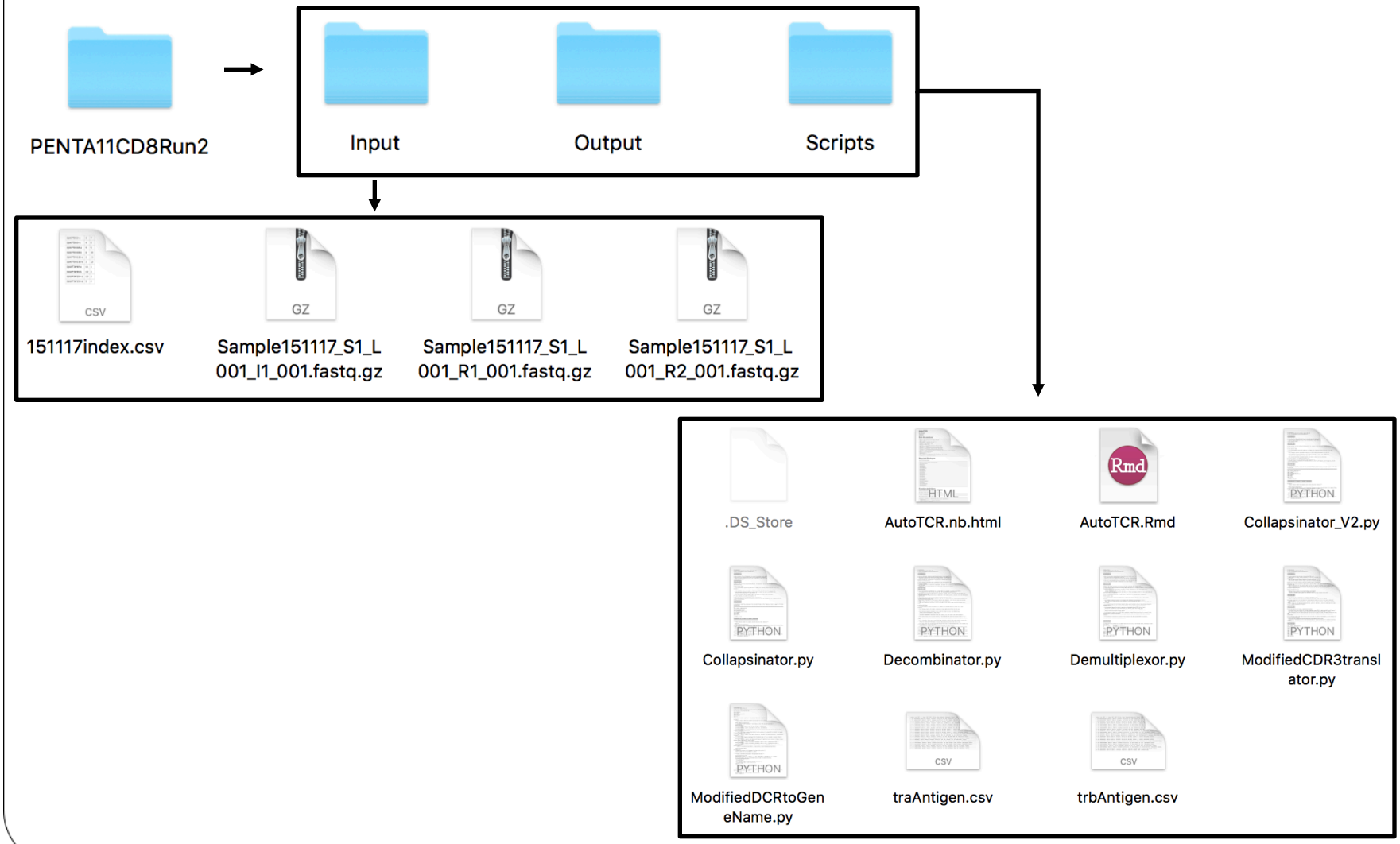
1. Create your file structure from template (Input, Output, Scripts)

2. Put your fastq files in ‘Input’

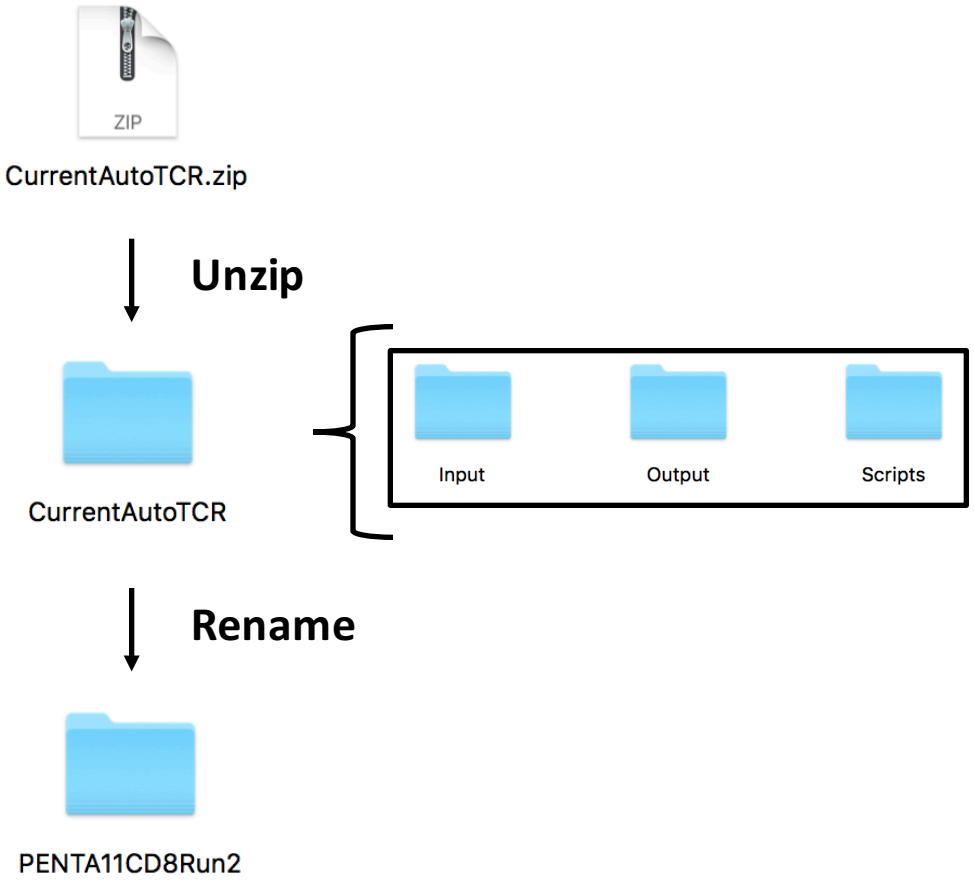
3. Create your index file (and put it in ‘Input’)

4. Open ‘AutoTCR.Rmd’ and edit the first section

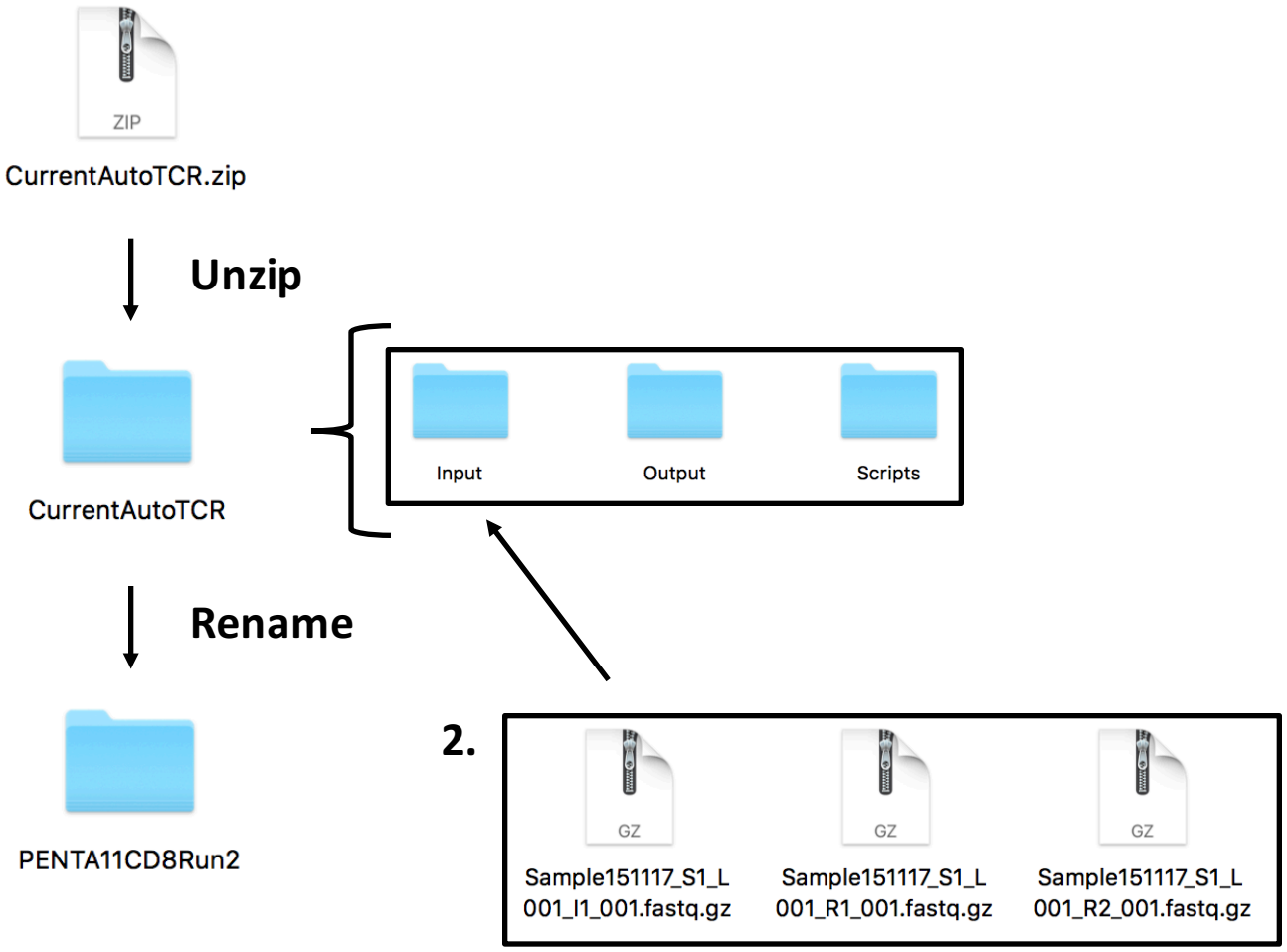
Automation & File Structure



1. Create File Structure From Template



1. Create File Structure From Template



3. The Index File

Sample Name	Index	
QUFCTW0-a	11	1
QUFCTW0-b	12	2
QUDCTW150-a	13	3
QUDCTW150-b	5	4
QUTPTIW48-a	4	5
QUTPTIW48-b	2	6
QUQPTIW48-a	3	7
QUQCTW150-a	2	11
QUQCTW150-b	3	12

QUFCTW0-a

→ ‘081217index.csv’

- ‘.csv’ – comma separated value

- Create with a text editor (i.e. bbedit)

```
1  QUFCTW0-a,11,1
2  QUFCTW0-b,12,2
3  QUDCTW150-a,13,3
4  QUDCTW150-b,5,4
5  QUTPTIW48-a,4,5
6  QUTPTIW48-b,2,6
7  QUQPTIW48-a,3,7
8  QUQCTW150-a,2,11
9  QUQCTW150-b,3,12
```

3. The Index File

Sample Name	Index	
QUFCTW0-a	11	1
QUFCTW0-b	12	2
QUDCTW150-a	13	3
QUDCTW150-b	5	4
QUTPTIW48-a	4	5
QUTPTIW48-b	2	6
QUQPTIW48-a	3	7
QUQCTW150-a	2	11
QUQCTW150-b	3	12

→ ‘081217index.csv’

- ‘.csv’ – comma separated value

- Create with a text editor (i.e. bbedit)

```
1  QUFCTW0-a,11,1
2  QUFCTW0-b,12,2
3  QUDCTW150-a,13,3
4  QUDCTW150-b,5,4
5  QUTPTIW48-a,4,5
6  QUTPTIW48-b,2,6
7  QUQPTIW48-a,3,7
8  QUQCTW150-a,2,11
9  QUQCTW150-b,3,12
```

QUFCTW0-a

QUFCTW0-a - Patient name

3. The Index File

Sample Name	Index	
QUFCTW0-a	11	1
QUFCTW0-b	12	2
QUDCTW150-a	13	3
QUDCTW150-b	5	4
QUTPTIW48-a	4	5
QUTPTIW48-b	2	6
QUQPTIW48-a	3	7
QUQCTW150-a	2	11
QUQCTW150-b	3	12

→ ‘081217index.csv’

- ‘.csv’ – comma separated value

- Create with a text editor (i.e. bbedit)

```
1  QUFCTW0-a,11,1
2  QUFCTW0-b,12,2
3  QUDCTW150-a,13,3
4  QUDCTW150-b,5,4
5  QUTPTIW48-a,4,5
6  QUTPTIW48-b,2,6
7  QUQPTIW48-a,3,7
8  QUQCTW150-a,2,11
9  QUQCTW150-b,3,12
```

QUFCTW0-a

QUFCTW0-a - Patient name

QUFCTW0-a - Time value (unit is irrelevant)

3. The Index File

Sample Name	Index	
QUFCTW0-a	11	1
QUFCTW0-b	12	2
QUDCTW150-a	13	3
QUDCTW150-b	5	4
QUTPTIW48-a	4	5
QUTPTIW48-b	2	6
QUQPTIW48-a	3	7
QUQCTW150-a	2	11
QUQCTW150-b	3	12

→ ‘081217index.csv’

- ‘.csv’ – comma separated value

- Create with a text editor (i.e. bbedit)

```
1  QUFCTW0-a,11,1
2  QUFCTW0-b,12,2
3  QUDCTW150-a,13,3
4  QUDCTW150-b,5,4
5  QUTPTIW48-a,4,5
6  QUTPTIW48-b,2,6
7  QUQPTIW48-a,3,7
8  QUQCTW150-a,2,11
9  QUQCTW150-b,3,12
```

QUFCTW0-a

QUFCTW0-a - Patient name

QUFCTW0-a - Time value (unit is irrelevant)

QUFCTW0-a - Chain (a = alpha, b = beta)

4. Edit AutoTCR

```
---
title: "AutoTCR"
author: "Ben Margetts"
date: "29/09/2017"
output:
  pdf_document: default
  html_notebook: default
  html_document: default
---

#Edit this section:

```{r}
#Run ID - what would you like to call this run?
runID <- 'pentarun5repeat'

#Email address to send the run report to
emailAddress <- 'ucbpmar@ucl.ac.uk'
sendEmail <- T #TRUE or FALSE

#Contains all data files
inputPath <- '~/Desktop/PENTA11Run5Repeat/Input'

#Path to write output to
outputPath <- '~/Desktop/PENTA11Run5Repeat/Output'

#Contains all decombinator scripts and this .Rmd script
scriptPath <- '~/Desktop/PENTA11Run5Repeat/Scripts'

#Contains a python installation with all prerequisite decombinator packages installed
python <- '~/anaconda2/bin/python'

#alpha and beta, or just 1?
numberChains <- 2

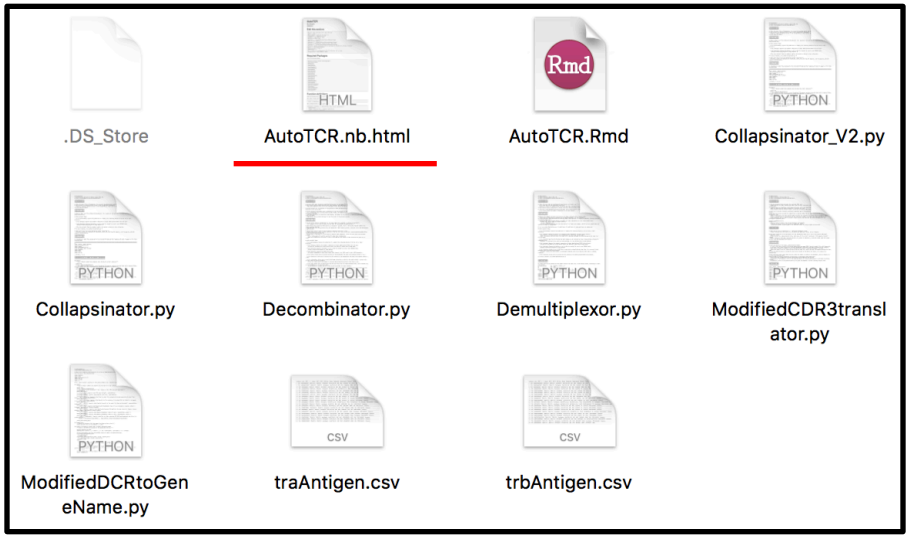
#Set to TRUE to run collapsinator with -N, otherwise, set to false.
collapsinatorN <- T #TRUE or FALSE
```
```

Cmd + S (Save file), Cmd + Alt + R (Run all sections)

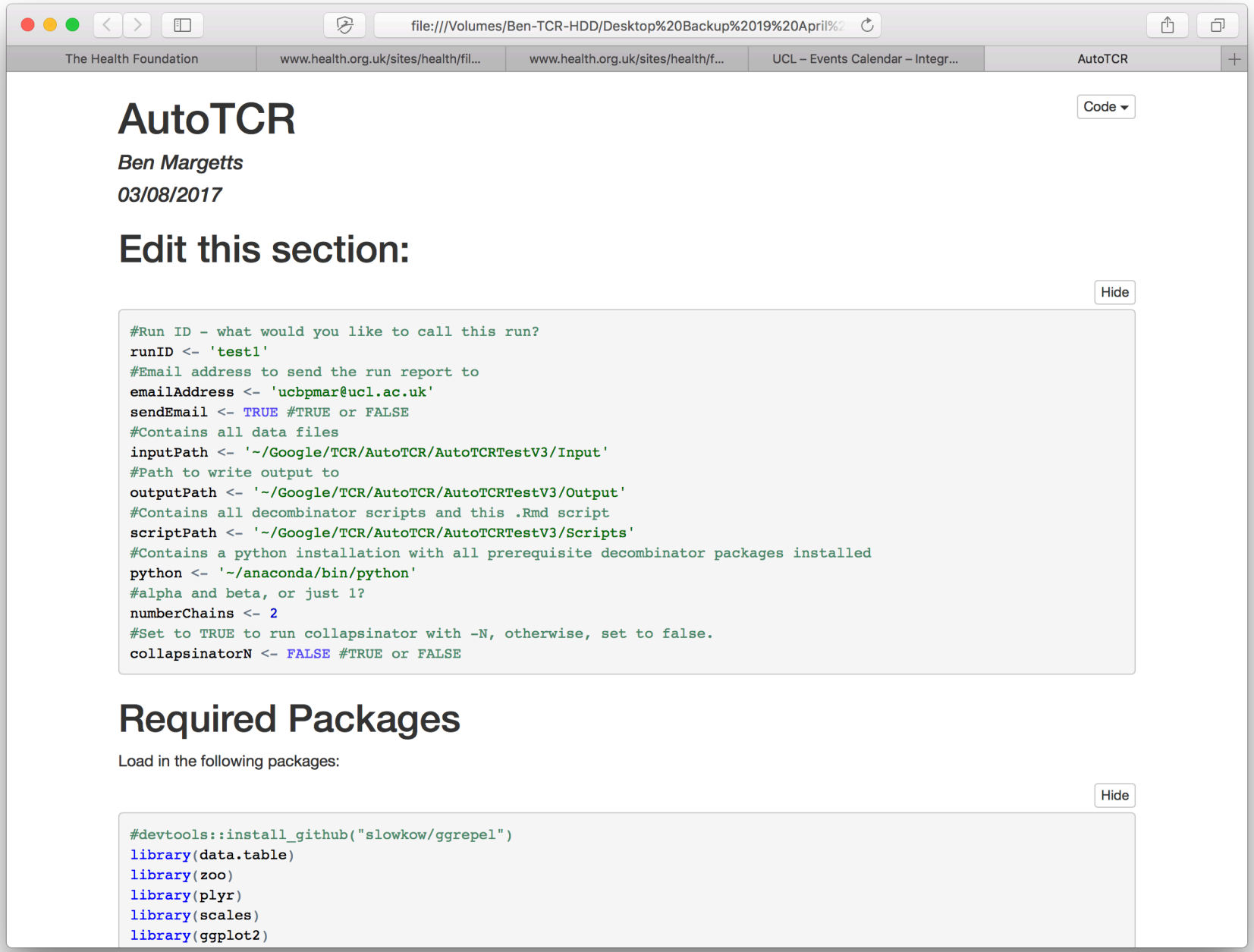
Reproducibility

Cmd + Alt + R (Run all sections)

- Produces a .html (web) file documenting the entire run and all output.
- All programmatic analysis is inherently reproducible. If the code is correct, it won't make a mistake



Reproducibility



Decombinator Pipeline

Decombinator: a tool for fast, efficient gene assignment in T-cell receptor sequences using a finite state machine

Niclas Thomas, James Heather, Wilfred Ndifon, John Shawe-Taylor, Benjamin Chain

Bioinformatics, Volume 29, Issue 5, 1 March 2013, Pages 542–550, https://doi.org/10.1093/bioinformatics/btt004

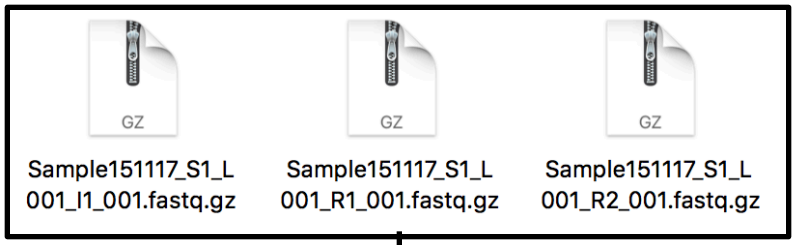
Published: 09 January 2013 **Article history**

Abstract

Summary: High-throughput sequencing provides an opportunity to analyse the repertoire of antigen-specific receptors with an unprecedented breadth and depth. However, the quantity of raw data produced by this technology requires efficient ways to categorize and store the output for subsequent analysis. To this end, we have defined a simple five-item identifier that uniquely and unambiguously defines each TcR sequence. We then describe a novel application of finite-state automaton to map Illumina short-read sequence data for individual TcRs to their respective identifier. An extension of the standard algorithm is also described, which allows for the presence of single-base pair mismatches arising from sequencing error. The software package, named Decombinator, is tested first on a set of artificial *in silico* sequences and then on a set of published human TcR-β sequences. Decombinator assigned sequences at a rate more than two orders of magnitude faster than that achieved by classical pairwise alignment algorithms, and with a high degree of accuracy (>88%), even after introducing up to 1% error rates in the *in silico* sequences. Analysis of the published sequence dataset highlighted the strong V and J usage bias observed in the human peripheral blood repertoire, which seems to be

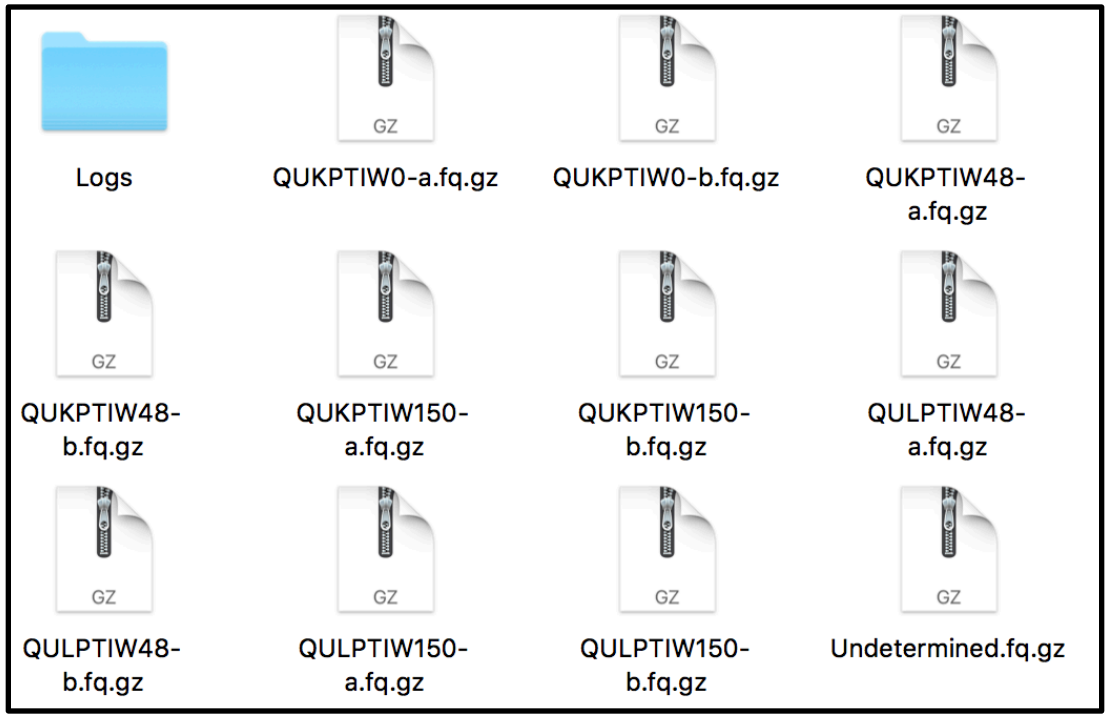
1. Demultiplexor
2. Decombinator
3. Collapsinator
4. CDR3 Translator
5. GN Translator

Demultiplexor



+

```
1  QUFCTW0-a,11,1
2  QUFCTW0-b,12,2
3  QUDCTW150-a,13,3
4  QUDCTW150-b,5,4
5  QUTPTIW48-a,4,5
6  QUTPTIW48-b,2,6
7  QUQPTIW48-a,3,7
8  QUQCTW150-a,2,11
9  QUQCTW150-b,3,12
```



Decombinator

QUKPTIW0-b.fq

```
@M01520:11:000000000-BHDMP:1:1101:13850:1643
TTCGTGATGTTCACGTCGTGATATAAACACTGACCATCAAGTGGTGGGAACACGTTTTTCAGGTCCTCCAGCACGGTCAGCCTGCTGCCGGCCCCGAAAGTCAGGACGTTGGCCCCAGCCCCCTCCCC
CGATGCTAAGCTGCTGGCACAGAGATACATGGCCGAGTCCCCCTGCTCTGTGCGCTGGATCTCCAAGTTGGAGAAAGATCCCTTAGGCCTCTCTGCAGAGAACCGATCACTGTGCAGCCTTTCTTTTT
CTATTTTAGCTTCTTTCTGTCATTT
+
>>>1>11BBFFF3F1FA1A00BAF33311BAAB1@1>A>3AF11FC100FABBBFGHGGHHB1GGHHHHGHHGHGGGGGFFHHGHHGFFG?E@/EE?ECGHHHHBHHGGGHAFGHG?>FFGGGCEAE/
C/<>//?2>?0?11F11?0?0?100111DF<00F?<@F<F<GA@.</<=0=00;-:--.;00000/90;0/0//;/;/B0F/;B/00/CF.9/0090/0.-/
99--;--99://;/;BA-9//;;FFF/9//9B/////:///;;/B///////
@M01520:11:000000000-BHDMP:1:1101:14014:1663
TTCGTGATGACCATTTCGTGATCAAGGCGATGACCATCAAGTGGTGGGAACACGTTTTTCAGGTCCTCTGTGACCGTGAGCCTGGTGCCCGGCCCGAAGTACTGCTCGTAGGAGCTCTGTCCCGGGCT
GCTGGCACAGAGGTACATAGATGTCTGGTTGGTGCTGGCGGACTCCAGAATCAGGGAGAAGCGCTCCTTCTTCTCTCTAGAGACACTGTACCCCTCGCCTTGATCACGACATGGTCATCACGACAGAT
CGGAAGAGCACACGTCTGAACTCCA
+
AAAAA?FFDBF11B1BFGF11BA3B11AE?AAAFFF>AAAAFFAGGCE0GFGBEGHHGGHHHCHHHHHGGFHHHHHGGGGGHHHHHHGHHHC/EEGGGGGGFHHHFHFAFEE/1FEHHHGGHFG?E/
>EFFHHF/FFHGHHEGHHDFGFFFBGDGHGHHCCGGHGD1<@-@CGFHFF0CDGD<0GA..CFGC@FCCCGHHHHHFGGF0099B/CBGFGF009CEF???GAFBFFFF?B=@BF-BFFB//FBF--;
9BF?B?-;-AFFAF-;AEF-//;F/F/
@M01520:11:000000000-BHDMP:1:1101:16754:1669
TTCGTGATAAAACCGTCGTGATGAGGACCTTGACCATCAAGTGGTGGGAACACCTTGTTCAGGTCCTCCTCACGACGGTTTTATCACGACAGATCGGAAGAGCACACGTCTGAACTCCAGGTCCTCAT
CACGACGGTTTTATCACGACAGATCGGAAGAGCACACGTCTGAACTCCAGTCACTCAAGTATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAATATTCTTTTCTTCTTCTTCTCTTATTCTTTCTTCT
TCATTACTTCTGTTCCTCCTCTTAT
+
3>AABBFFFF5FGCGGFGEGFGHHHHHHHHDAFCFFAAAABF11FGAEAGFGFEGHHHHHHHHBFFGHH0FHHHHGGCGEGGEHHHHHGFG/
EFGHHFCGGFHH1FHHGHHHGHHHHHHHHHGHGHHHHHHFHGGGEGGGGEHHHHHGCGGGHHHHGGGGHHHHHHHGHHHG1FHHHGHHHHEHFHFEDGFHFGF<ECHFEH-:@GCCH:G;G0.:00:9-
--.0;00///;/;;//:/B;//9/////9//;/9////////;/////9//////;-///;
@M01520:11:000000000-BHDMP:1:1101:13872:1674
TTCGTGATATAAATGTCGTTGATAACATAATGACCATCAAGTGGTGGGAACACGTTTTTCAGGTCCTCTGTGACCGTGAGCCTGGTGCCCGGCCCGAAGTACTGCTCGTAGGAGCTCTGTCCCGGGCT
GCTGGCACAGAGGTACATAGATGTCTGGTTGGTGCTGGCGGACTCCAGAATCAGGGAGAAGCGCTCCTTCTTCTCTCTAGAGACACTGTACCCCTCAGGAATATCTCCTTTTTCTTTCATTTTAACAT
CATTTTGTTATCAACGACATTTATT
+
1>>1AA1CFFDBFGGGGGGAF1FADDGGGBB@?
FFBAA>AAFGCGFEGAGFGBEGHHGGHHH3FGHHHHHHHHHHHGGGGGHHHHHHHHHHGGGEEGGCEFEHHHGHHGGGGGHHFHHHHHHHHEGGGDFEHHGGHHHHHHGHHHHHFHDHHHHHHGGHDGG
GHHEHGGCCDGHHHEHHHHFFFEFDDGGFGBGGGGGGGGGGFGGGBBC9FCFGGBFBFGGEAD?/FF.B/FFFFFFFFFFFFFFBFFBFF/99FF/FF/;/BBBFB//:..9-;FF/B/
@M01520:11:000000000-BHDMP:1:1101:17746:1677
TTCGTGATCACACAGTCGTGATCCTAACGCTGACCATCAAGTGGTGGGAACACGTTTTTCAGGTCCTCTGGTCAATATCTGATGGGATGGTCCACATTTCCTCCTCCCCTGGGGCAGGAGGGTGTGGT
AGCTTGTCCAGGATGGCTGAAATCCCATGGAAGTTGATGGTCATGGGATTAACCCTTTCTTCTCTCAGAGTGTCCAGCAACCCTGCCTGGCCAGGATTGGAATCCGTCCACAGTTAAGCTTTGCGTTA
GGATCACGACTGTGTGATCACGACA
+
1>>1>AFFFFFFGGAGFEF1FFFHHCFFGGFFFFFFAAAABFGFG1FGAGFGBEGHHGGHHGFHHHHHHHHHHHHHHHHHHHHHHHHHHHGGHHHHHHHHHHFHHHHHHGGHGGGGGGGGFGGGEFGGGH
HGHHHGHHHHHHHHHGHHGHHHHHHHHHHHGHHHHHHEHHHHHHFHHHHHHHHHHHHHHHEFGHHGHGGFHFFHHGHHHGHHHHHHGHEHHHGGFHHHGGHGGHHGHHHHHGGGGGGGGGG
GGEGGGGB?AGGGGGFFFFFFBB@;
@M01520:11:000000000-BHDMP:1:1101:12877:1678
TTCGTGATGCATTTGTCGTGATAACCACGCTGACCATCAAGTGGTGGGAACACGTTTTTCAGGTCCTCTGGAAAGGGAAGAGGGGTTGGAGCCAGGGTTGCTCTGAGAGCTGTCTGGTTCTGGTAGGG
GCTCTGTGTGTGTGAGAGAGAGCCCTGGGAATGGGCCGTAATGAAGCACAACTATGTCATTCTAGGCGTGGTTATCACGACAAATGCATCACGACAGATCGGAAGAGCACACGTCTGAACTCCAGTCA
CTCAAGTATCTCGTATGCCGTCTTC
+
1>A1AABFFFFFGGBGGGFFCGGGHCFGGGBBBFFF>AAABFGFGCFGGGFGBFFGHGGHHHHHHHHHHHHHHHBGHGHHGHHHGGGGGGHHHGGHHFHHGHHHHHHHHHHHHHHHHHGHHGFHHHHEE
GGGGHHHHHHHHHHHHFGEFGGHHGHHHGHGHHHHHHGGGCGHFHHHHHGHGHHHHHHGHHHHHHFHGGFCGFFHFHGGGGGGFHHHHHGHHGGCCGGGGGGBGGGGGGGGG?
```

Looks for rearranged TCRs

43, 11, 0, 2,
ATCGGGGGAGGGGG,
M01520:11:000000000-
BHDMP:1:1101:13850:1643,

V index, J index, # V deletions, # J deletions, insert, ID

Collapsinator

Productive sequences? **Not PCR amplified?**

43, 11, 0, 2,
ATCGGGGGAGGGGG,
M01520:11:000000000-
BHDMP:1:1101:13850:1643,

43, 11, 0, 2,
ATCGGGGGAGGGGG,
M01520:11:000000000-
BHDMP:1:1101:13850:1643,

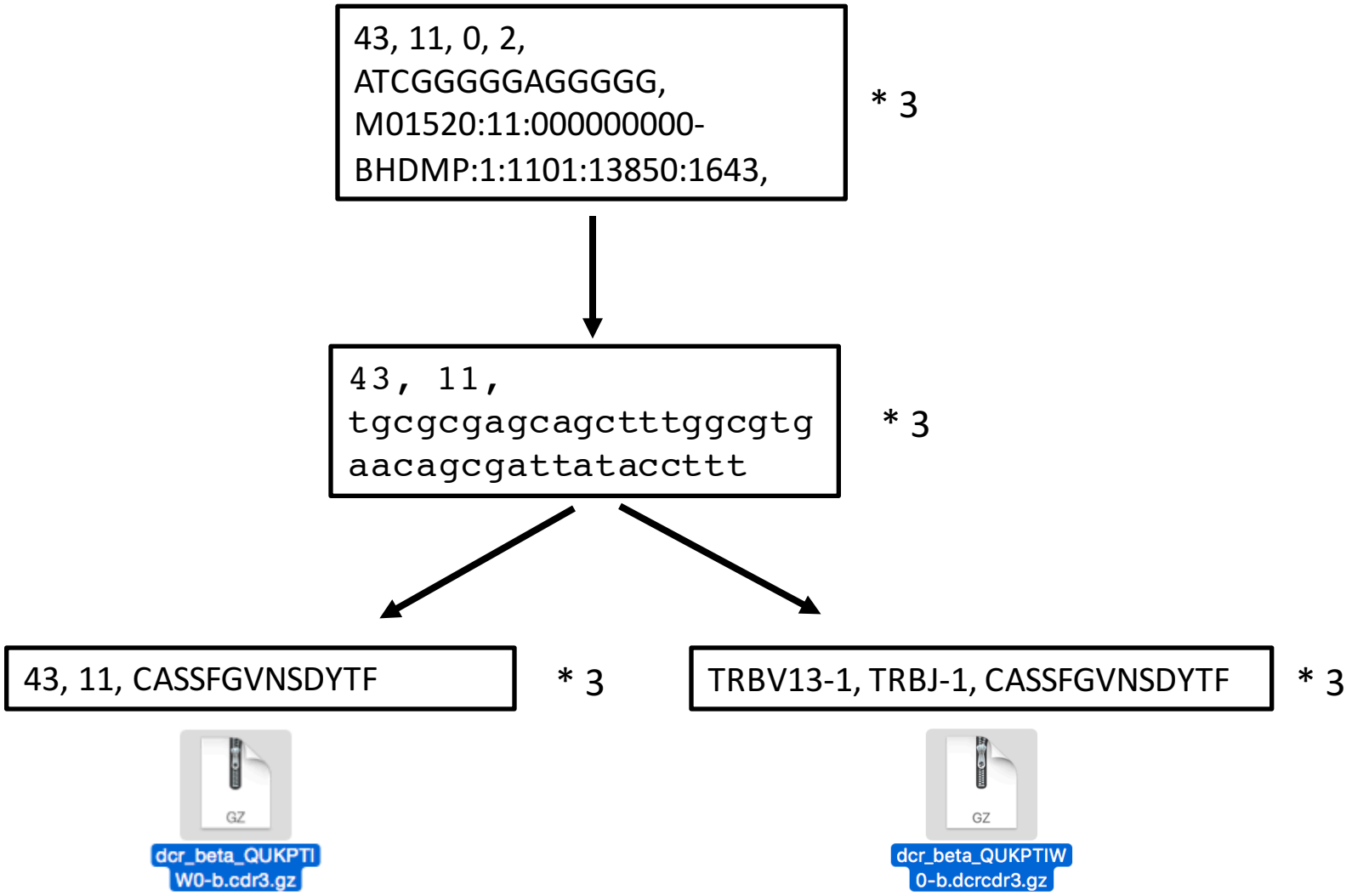
43, 11, 0, 2,
ATCGGGGGAGGGGG,
M01520:11:000000000-
BHDMP:1:1101:13850:1643,

43, 11, 0, 2,
ATCGGGGGAGGGGG,
M01520:11:000000000-
BHDMP:1:1101:13850:1643,

* 3

dcr_beta_QUKPTIW0-b.freq.gz

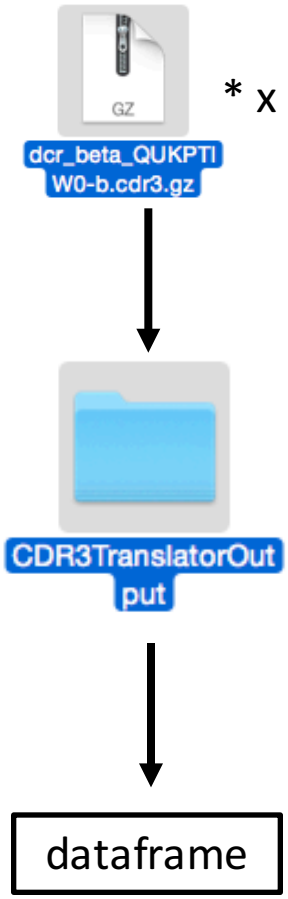
CDR3 Translator & GN Translator



Building a Dataframe

Key elements to capture:

- CDR3 Sequence
- Chain (alpha/beta)
- ID
- Frequency
- Additional computations (for each sequence)



Building a Dataframe

CAAIRGAGSYQLTF	1
CAAIRQAGTALIF	1
CAAIVAGTALIF	1
CAAIVIPGSQGNLIF	1
CAALWGYSSASKIIF	1
CAAMDSNMDSNYQLIW	1
CAAPHRCGNEKLTF	11
CAAPHRFGNEKLTF	209

QUKPTIW0-b = Patient name

\+ QUKPTIW0-b = Time value

QUKPTIW0-b = Chain

\+

CAAPTPLYNTDKLIF	2
CAARETGTASKLTF	1
CAARETSGSRLTF	1
CAARKAETRGSRLTF	1
CAARKAETSGSRLTF	103
CAARKTSGSRLTF	1
CAARRAETSGSRLTF	1
CAARVRDMRF	12

QUKPTIW0-a = Patient name

\+ QUKPTIW0-a = Time value

QUKPTIW0-a = Chain

=

dataframe

Building a Dataframe

dataframe →

TTAnalysis.Rmd × dat ×

Filter

	ID	month	chain	V1	V2
1	A	0	alpha	CAADQTGANNLFF	1
2	A	0	alpha	CAADSGGYQKVTF	1
3	A	0	alpha	CAADSGYSTLTF	27
4	A	0	alpha	CAADTGRRALTF	6
5	A	0	alpha	CAAFSGGYNKLIF	2
6	A	0	alpha	CAAIRGAGSYQLTF	1
7	A	0	alpha	CAAIRQAGTALIF	1
8	A	0	alpha	CAAIVAGTALIF	1
9	A	0	alpha	CAAIVIPGSQGNLIF	1
10	A	0	alpha	CAALWGYSSASKIIF	1
11	A	0	alpha	CAAMDSNMDSNYQLIW	1
12	A	0	alpha	CAAPHRCGNEKLTF	11
13	A	0	alpha	CAAPHRFGNEKLTF	209
14	A	0	alpha	CAAPHRVGNEKLTF	1
15	A	0	alpha	CAAPPRFGNEKLTF	1
16	A	0	alpha	CAAPRGYSTLTF	1
17	A	0	alpha	CAAPTPLYNTDKLIF	2
18	A	0	alpha	CAARETGTASKLTF	1
19	A	0	alpha	CAARETSGSRLTF	1
20	A	0	alpha	CAARKAETRGSRLTF	1
21	A	0	alpha	CAARKAETSGSRLTF	103
22	A	0	alpha	CAARKTSGSRLTF	1
23	A	0	alpha	CAARRAETSGSRLTF	1
24	A	0	alpha	CAARVRDMRF	12
25	A	0	alpha	CAASAGGKLIF	1
26	A	0	alpha	CAASALPGNTPLVF	13
27	A	0	alpha	CAASDGNKLVF	1
28	A	0	alpha	CAASEERGATNKLIF	1
29	A	0	alpha	CAASFGYKLSF	2
30	A	0	alpha	CAASGDTDKLIF	1
31	A	0	alpha	CAASGGAANKLIF	1

Showing 1 to 31 of 121,406 entries

Building a Dataframe

dataframe →

TTAnalysis.Rmd × dat ×

Filter

	ID	month	chain	V1	V2	CDR3Len
1	A	0	alpha	CAADQTGANNLFF	1	13
2	A	0	alpha	CAADSGGYQKVTF	1	13
3	A	0	alpha	CAADSGYSTLTF	27	12
4	A	0	alpha	CAADTGRRALTF	6	12
5	A	0	alpha	CAAFSGGYNKLIF	2	13
6	A	0	alpha	CAAIRGAGSYQLTF	1	14
7	A	0	alpha	CAAIRQAGTALIF	1	13
8	A	0	alpha	CAAIVAGTALIF	1	12
9	A	0	alpha	CAAIVIPGSQGNLIF	1	15
10	A	0	alpha	CAALWGYSSASKIIF	1	15
11	A	0	alpha	CAAMDSNMDSNYQLIW	1	16
12	A	0	alpha	CAAPHRCGNEKLTF	11	14
13	A	0	alpha	CAAPHRFGNEKLTF	209	14
14	A	0	alpha	CAAPHRVGNEKLTF	1	14
15	A	0	alpha	CAAPPRFGNEKLTF	1	14
16	A	0	alpha	CAAPRGYSTLTF	1	12
17	A	0	alpha	CAAPTPLYNTDKLIF	2	15
18	A	0	alpha	CAARETGTASKLTF	1	14
19	A	0	alpha	CAARETSGSRLTF	1	13
20	A	0	alpha	CAARKAETRGSRLTF	1	15
21	A	0	alpha	CAARKAETSGSRLTF	103	15
22	A	0	alpha	CAARKTSGSRLTF	1	13
23	A	0	alpha	CAARRAETSGSRLTF	1	15
24	A	0	alpha	CAARVRDMRF	12	10
25	A	0	alpha	CAASAGGKLIF	1	11
26	A	0	alpha	CAASALPGNTPLVF	13	14
27	A	0	alpha	CAASDGNKLVF	1	11
28	A	0	alpha	CAASEERGATNKLIF	1	15
29	A	0	alpha	CAASFGYKLSF	2	11
30	A	0	alpha	CAASGDTDKLIF	1	12
31	A	0	alpha	CAASGGAANKLIF	1	13

Showing 1 to 31 of 121,406 entries

Building a Dataframe

dataframe →

TTAnalysis.Rmd × dat ×

Filter

	ID	month	chain	V1	V2	CDR3Len	normV2
1	A	0	alpha	CAADQTGANNLFF	1	13	0.01476015
2	A	0	alpha	CAADSGGYQKVTF	1	13	0.01476015
3	A	0	alpha	CAADSGYSTLTF	27	12	0.39852399
4	A	0	alpha	CAADTGRRALTF	6	12	0.08856089
5	A	0	alpha	CAAFSGGYNKLIF	2	13	0.02952030
6	A	0	alpha	CAAIRGAGSYQLTF	1	14	0.01476015
7	A	0	alpha	CAAIRQAGTALIF	1	13	0.01476015
8	A	0	alpha	CAAIVAGTALIF	1	12	0.01476015
9	A	0	alpha	CAAIVIPGSQGNLIF	1	15	0.01476015
10	A	0	alpha	CAALWGYSSASKIIF	1	15	0.01476015
11	A	0	alpha	CAAMDSNMDSNYQLIW	1	16	0.01476015
12	A	0	alpha	CAAPHRCGNEKLTF	11	14	0.16236162
13	A	0	alpha	CAAPHRFGNEKLTF	209	14	3.08487085
14	A	0	alpha	CAAPHRVGNEKLTF	1	14	0.01476015
15	A	0	alpha	CAAPPRFGNEKLTF	1	14	0.01476015
16	A	0	alpha	CAAPRGYSTLTF	1	12	0.01476015
17	A	0	alpha	CAAPTPLYNTDKLIF	2	15	0.02952030
18	A	0	alpha	CAARETGTASKLTF	1	14	0.01476015
19	A	0	alpha	CAARETSGSRLTF	1	13	0.01476015
20	A	0	alpha	CAARKAETRGSRLTF	1	15	0.01476015
21	A	0	alpha	CAARKAETSGSRLTF	103	15	1.52029520
22	A	0	alpha	CAARKTSGSRLTF	1	13	0.01476015
23	A	0	alpha	CAARRAETSGSRLTF	1	15	0.01476015
24	A	0	alpha	CAARVRDMRF	12	10	0.17712177
25	A	0	alpha	CAASAGGKLIF	1	11	0.01476015
26	A	0	alpha	CAASALPGNTPLVF	13	14	0.19188192
27	A	0	alpha	CAASDGNKLVF	1	11	0.01476015
28	A	0	alpha	CAASEERGATNKLIF	1	15	0.01476015
29	A	0	alpha	CAASFGYKLSF	2	11	0.02952030
30	A	0	alpha	CAASGDTDKLIF	1	12	0.01476015
31	A	0	alpha	CAASGGAANKLIF	1	13	0.01476015

Showing 1 to 31 of 121,406 entries

Building a Dataframe

TTAnalysis.Rmd × dat ×

Filter

	ID	month	chain	V1	V2	CDR3Len	normV2
1	A	0	alpha	CAADQTGANNLFF	1	13	0.01476015
2	A	0	alpha	CAADSGGYQKVTF	1	13	0.01476015
3	A	0	alpha	CAADSGYSTLTF	27	12	0.39852399
4	A	0	alpha	CAADTGRRALTF	6	12	0.08856089
5	A	0	alpha	CAAFSGGYNKLIF	2	13	0.02952030
6	A	0	alpha	CAAIRGAGSYQLTF	1	14	0.01476015
7	A	0	alpha	CAAIRQAGTALIF	1	13	0.01476015
8	A	0	alpha	CAAIVAGTALIF	1	12	0.01476015
9	A	0	alpha	CAAIVIPGSQGNLIF	1	15	0.01476015
10	A	0	alpha	CAALWGYSSASKIIF	1	15	0.01476015
11	A	0	alpha	CAAMDSNMDSNYQLIW	1	16	0.01476015
12	A	0	alpha	CAAPHRCGNEKLTF	11	14	0.16236162
13	A	0	alpha	CAAPHRFGNEKLTF	209	14	3.08487085
14	A	0	alpha	CAAPHRVGNEKLTF	1	14	0.01476015
15	A	0	alpha	CAAPPRFGNEKLTF	1	14	0.01476015
16	A	0	alpha	CAAPRGYSTLTF	1	12	0.01476015
17	A	0	alpha	CAAPTPLYNTDKLIF	2	15	0.02952030
18	A	0	alpha	CAARETGTASKLTF	1	14	0.01476015
19	A	0	alpha	CAARETSGSRLTF	1	13	0.01476015
20	A	0	alpha	CAARKAETRGSRLTF	1	15	0.01476015
21	A	0	alpha	CAARKAETSGSRLTF	103	15	1.52029520
22	A	0	alpha	CAARKTSGSRLTF	1	13	0.01476015
23	A	0	alpha	CAARRAETSGSRLTF	1	15	0.01476015
24	A	0	alpha	CAARVRDMRF	12	10	0.17712177
25	A	0	alpha	CAASAGGKLIF	1	11	0.01476015
26	A	0	alpha	CAASALPGNTPLVF	13	14	0.19188192
27	A	0	alpha	CAASDGNKLVF	1	11	0.01476015
28	A	0	alpha	CAASEERGATNKLIF	1	15	0.01476015
29	A	0	alpha	CAASFGYKLSF	2	11	0.02952030
30	A	0	alpha	CAASGDTDKLIF	1	12	0.01476015
31	A	0	alpha	CAASGGAANKLIF	1	13	0.01476015

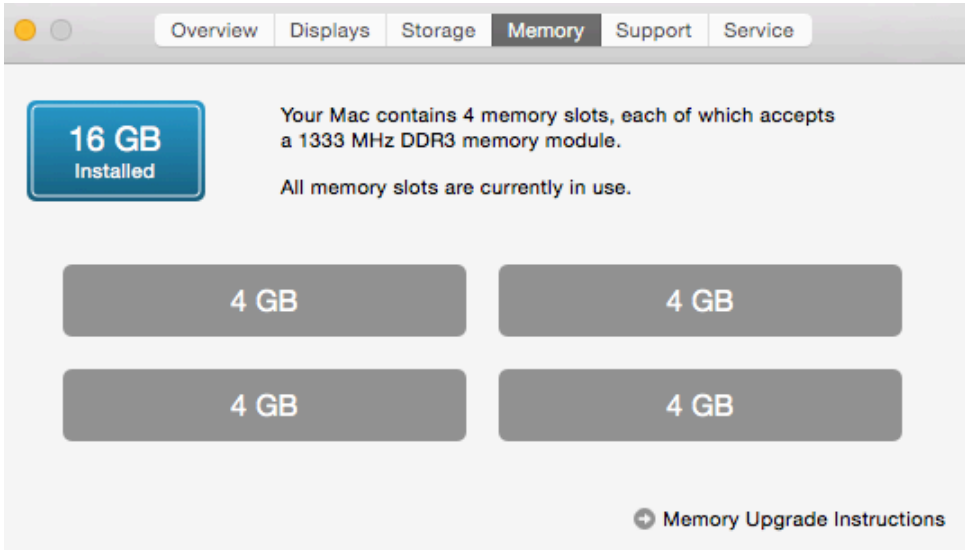
Showing 1 to 31 of 121,406 entries

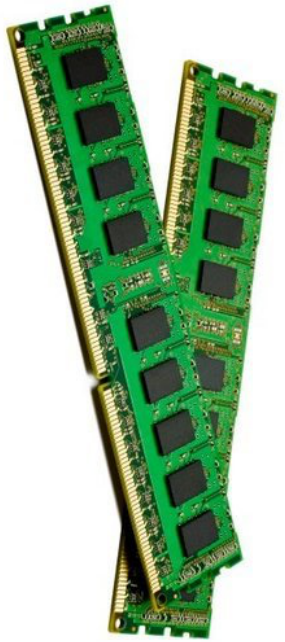
+ Your analysis goes here!

Why this format?

Showing 1 to 31 of 121,406 entries

- Your data is loaded into RAM (fast memory)
- Your computer has 4 – 32 GB of available RAM





Why this format?

- Every **character** in the dataframe **takes up some memory**

- We need to **condense** the **information** as much as possible to hold and analyse this much data. If something takes 1 hour to run. 10 * as much data will take 10 hours.

In a .csv file:

CAADQTGANNLFF, 250	17 bytes	58,823,529 copies
CAADQTGANNLFF, CAADQTGANNLFF, CAADQTGANNLFF, . . . CAADQTGANNLFF	4.25 kilobytes	235,294 copies

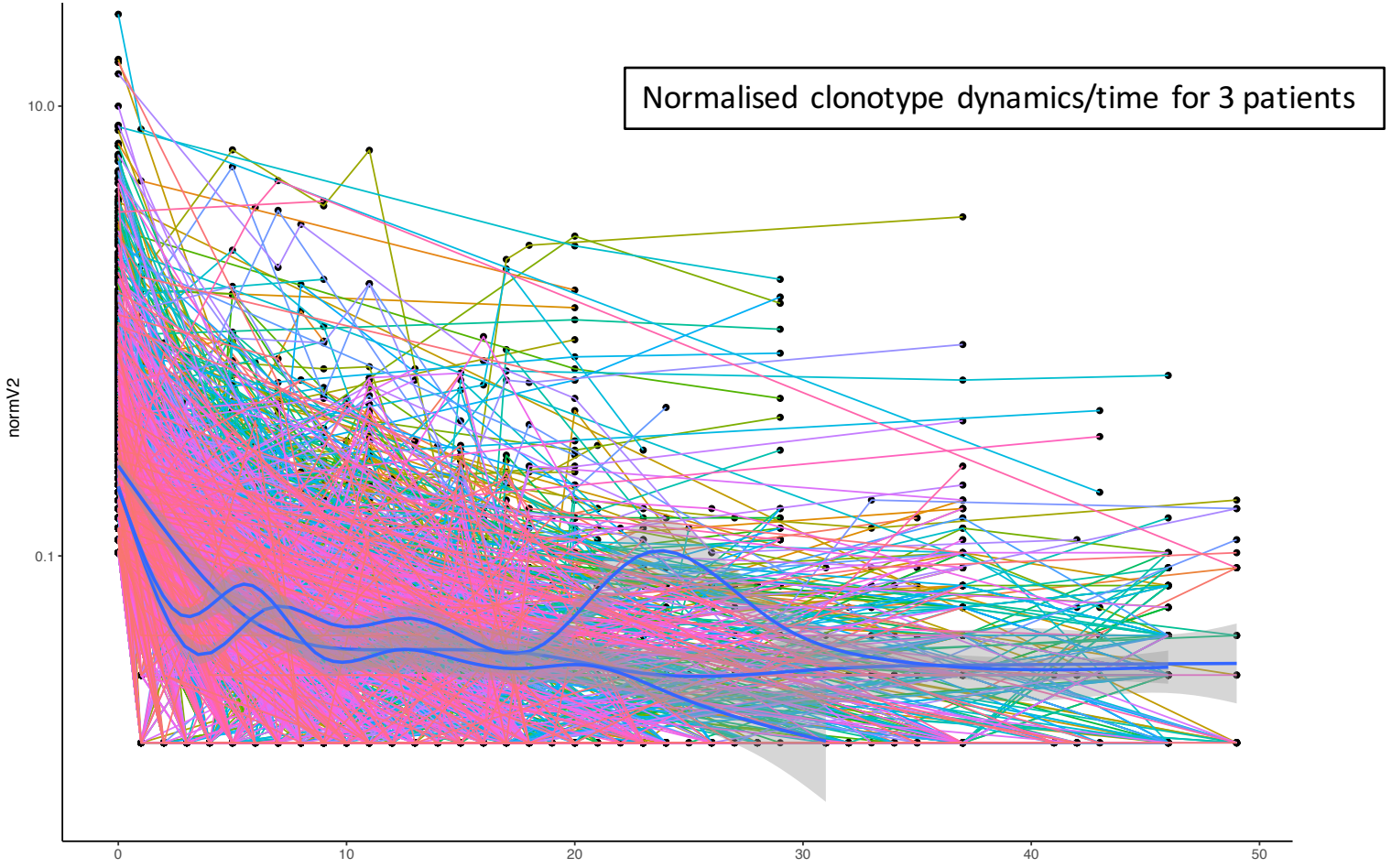
Automated Analysis

Q. How do we know the pipeline works?
A. The data are consistent with our biological understanding.

Automated Analysis

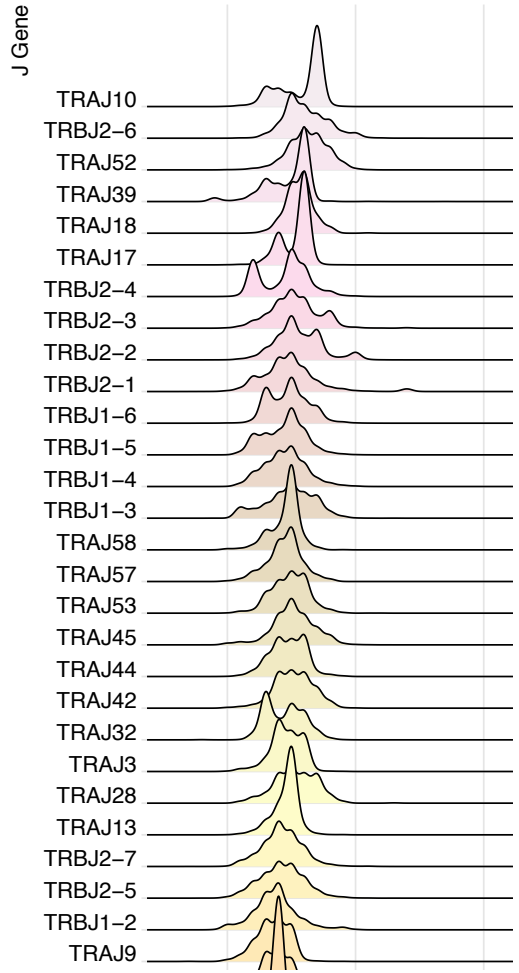
Q. How do we know the pipeline works?

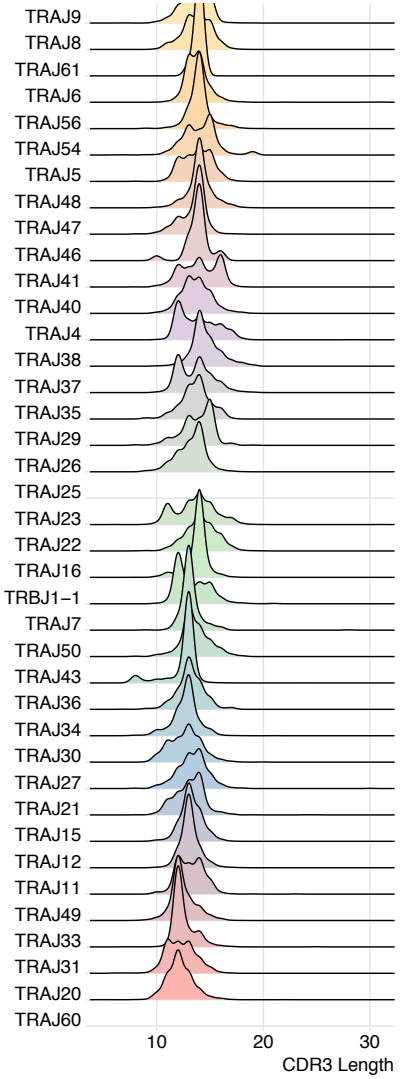
A. The data are consistent with our biological understanding.



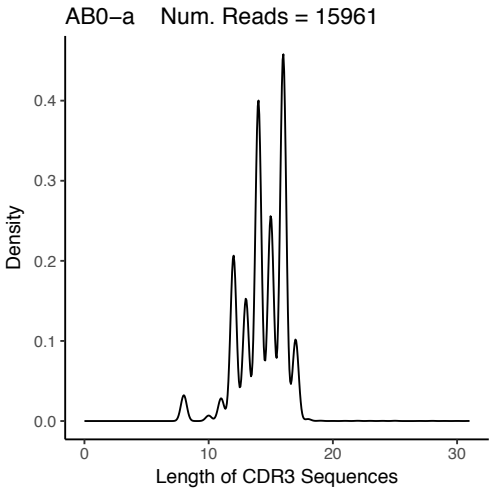
Automated Analysis

Using the data structure presented previously, we have automated:





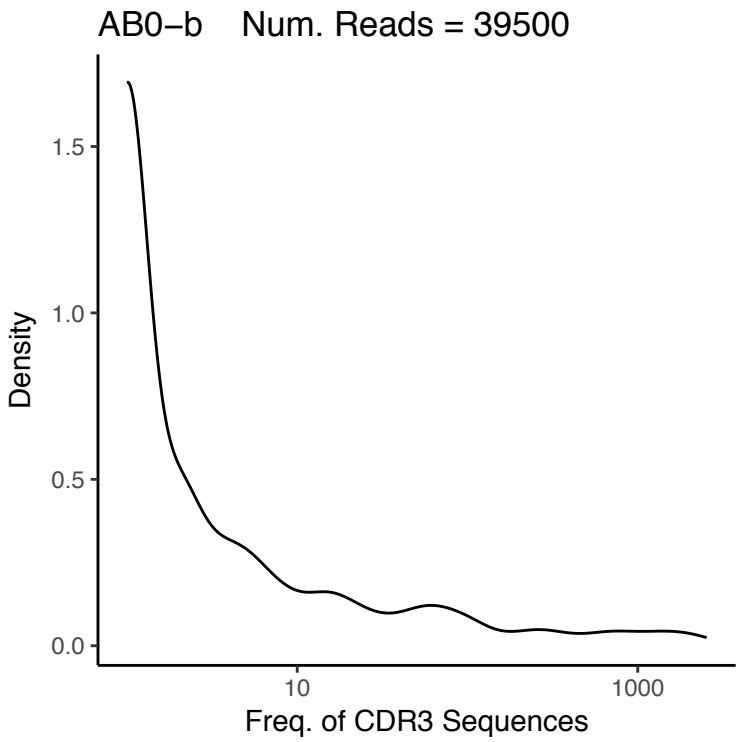
Spectratyping

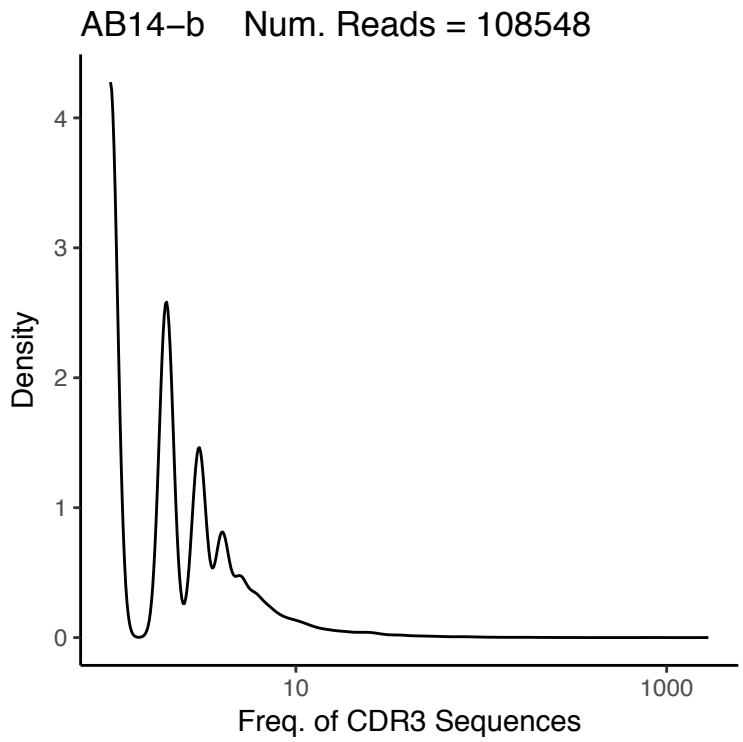


Automated Analysis

Using the data structure presented previously, we have automated:

Clonotype Frequency Density

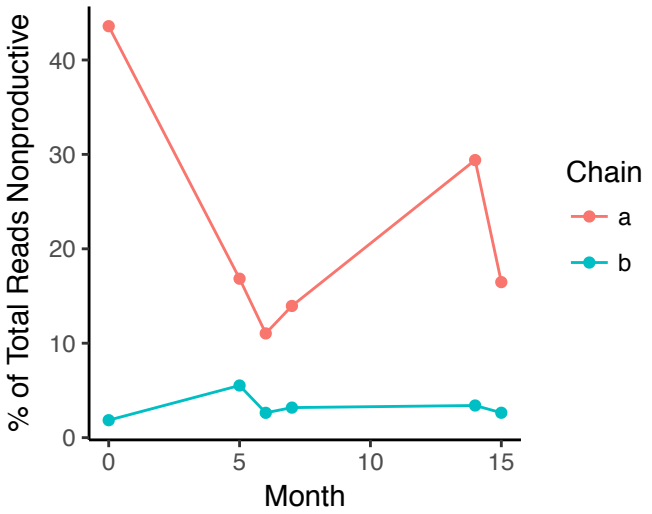


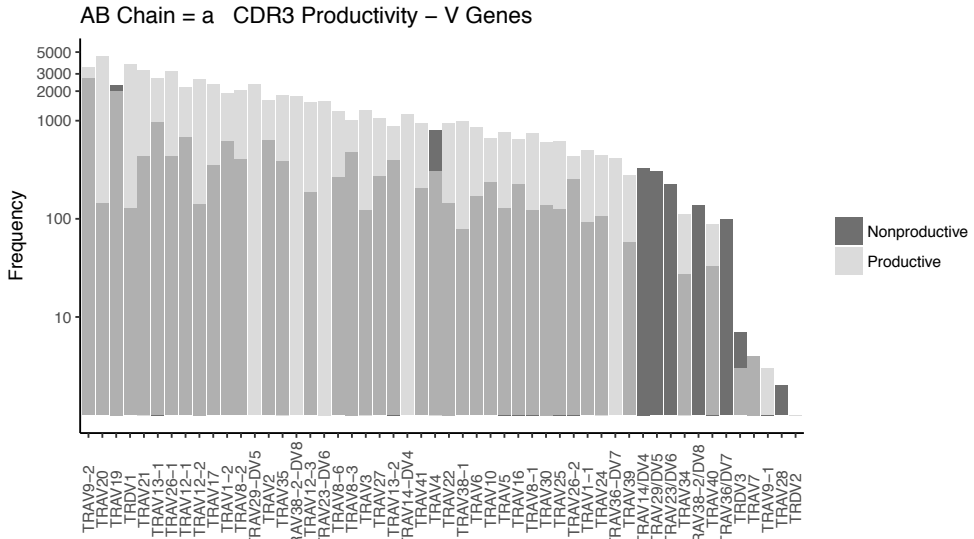


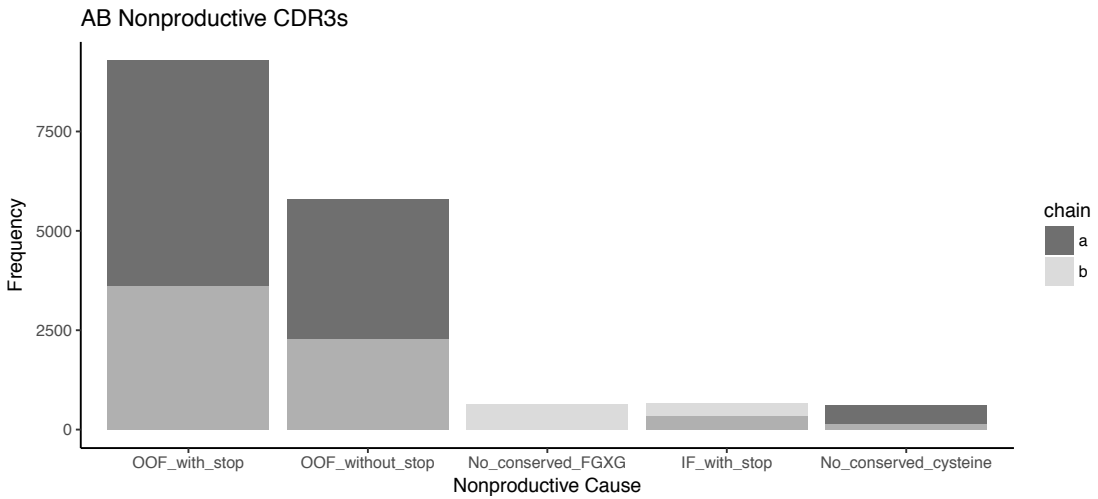
Automated Analysis

Using the data structure presented previously, we have automated:

Nonproductive analysis



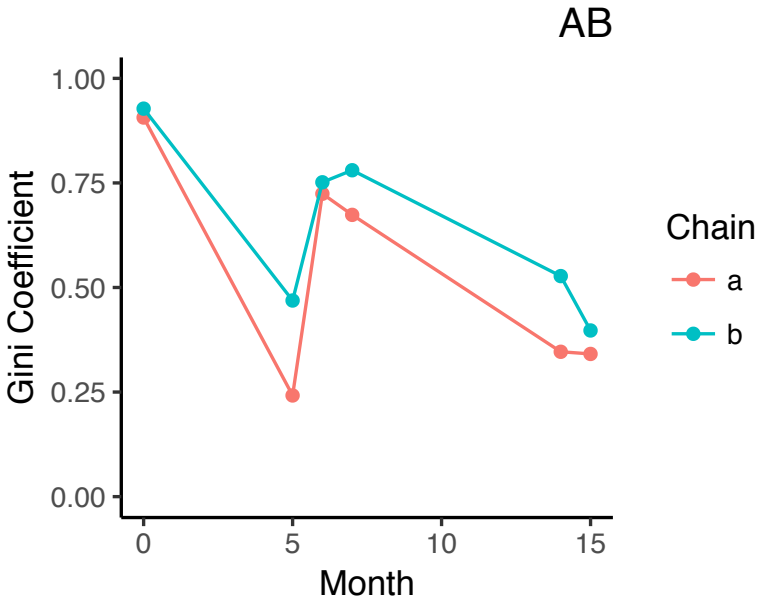


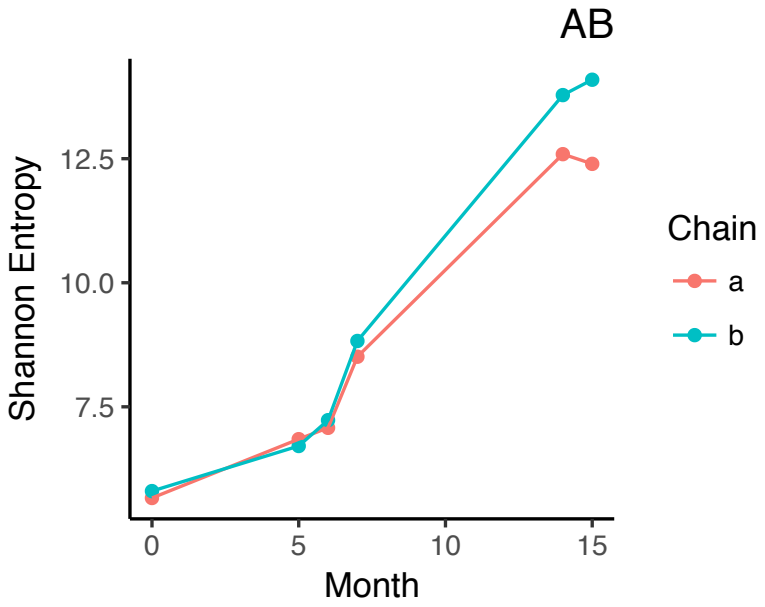


Automated Analysis

Using the data structure presented previously, we have automated:

Diversity Analysis

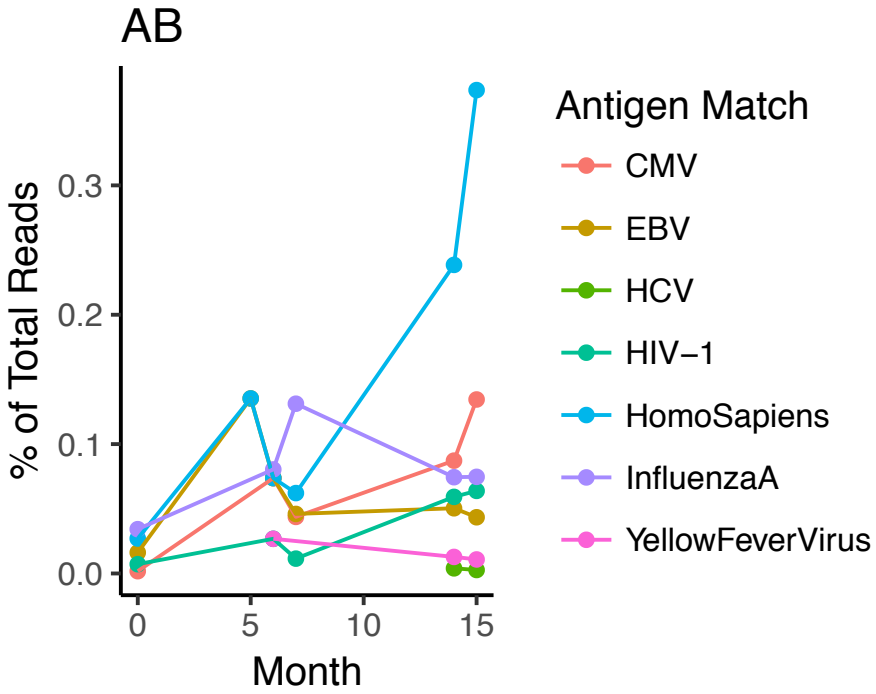


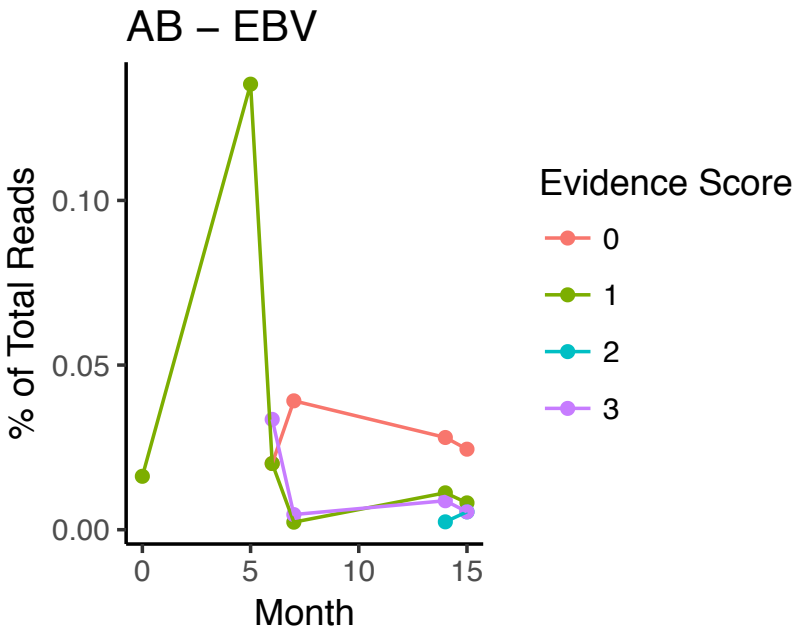


Automated Analysis

Using the data structure presented previously, we have automated:

Antigen Specificity Database Matching
(Although we don't recommend it!)

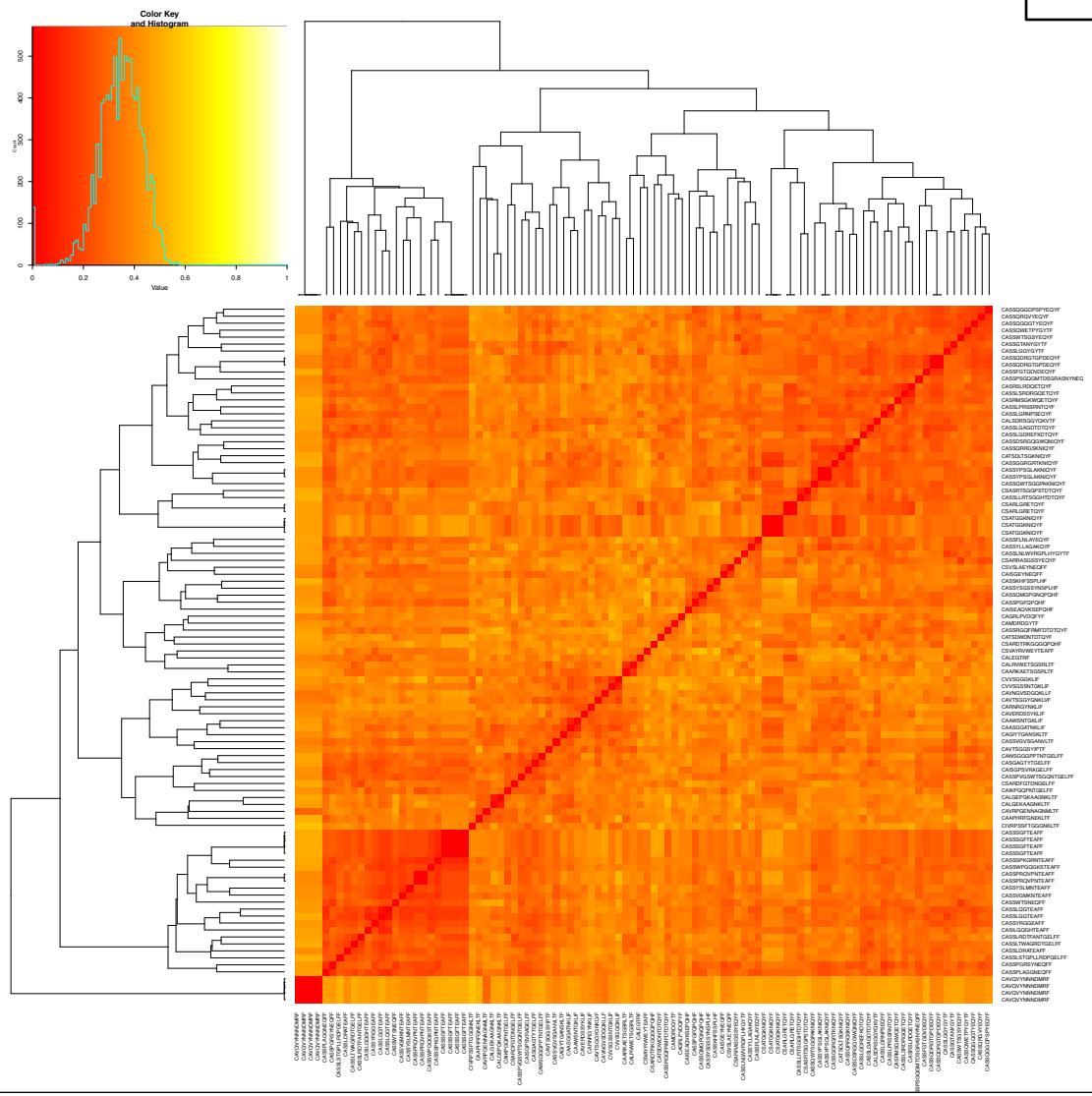


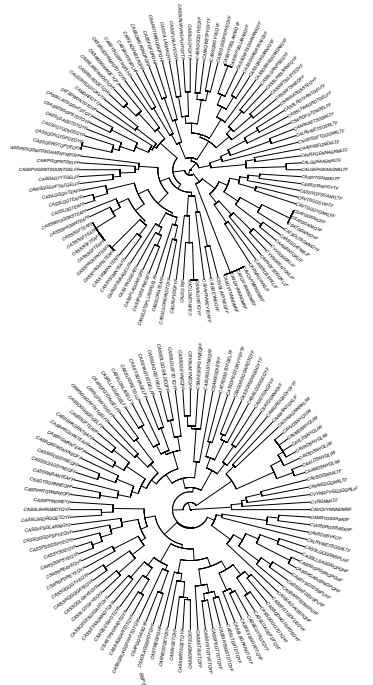


Automated Analysis

Using the data structure presented previously, we have automated:

Clonotype Similarity Analyses

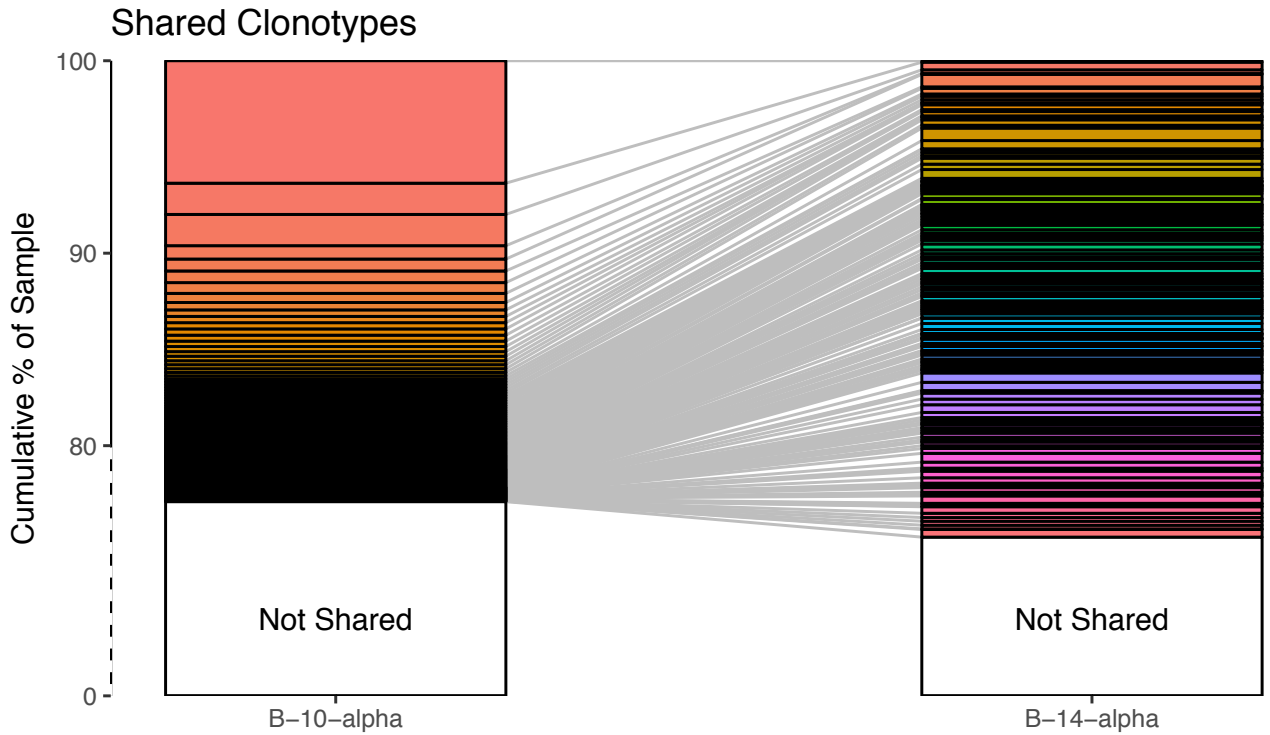




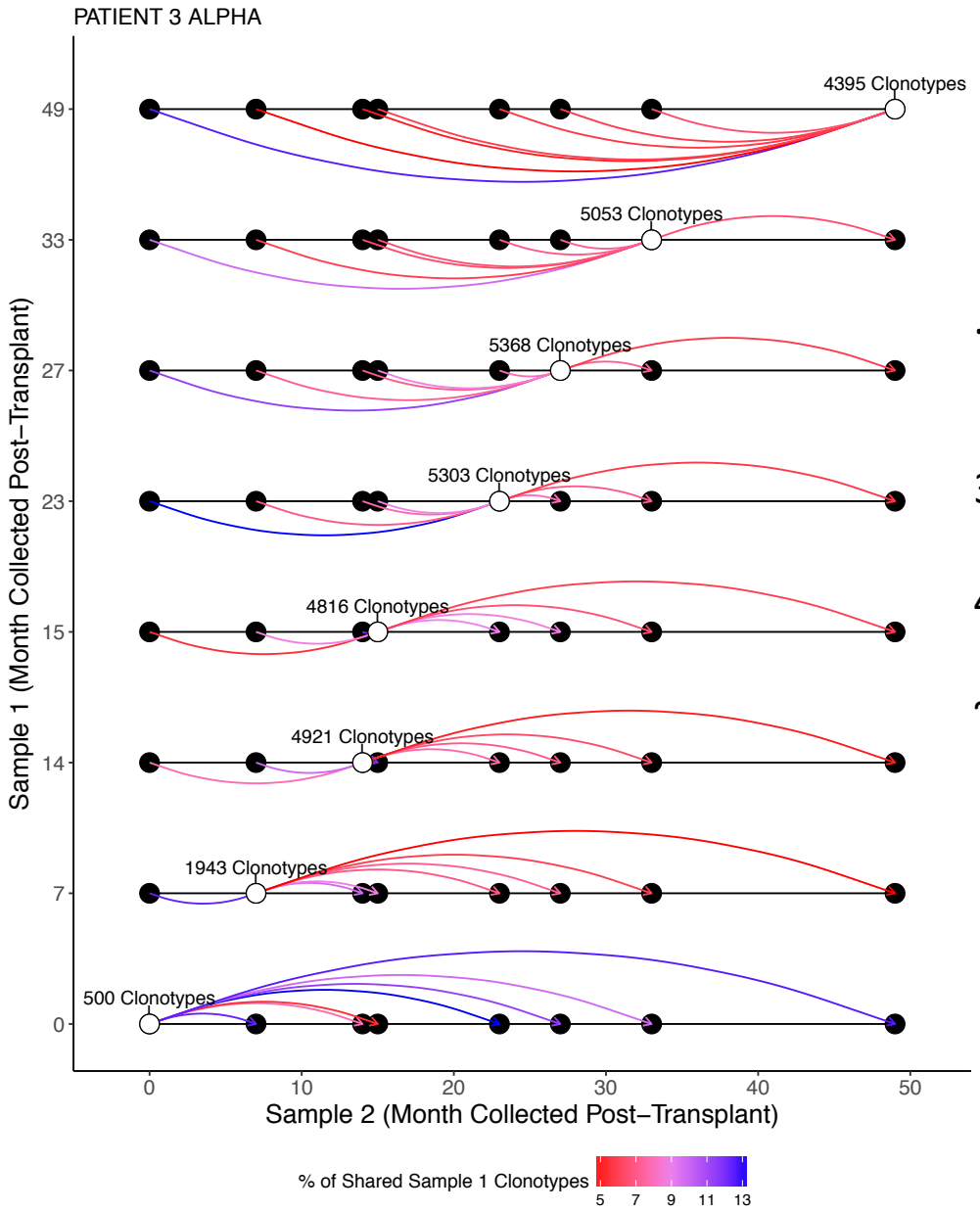
Making Your Own Analysis

- All the **code** is **open source** and available in the script.
- Custom analysis is highly encouraged.

- Draw the plot with pen and paper, break it down into manageable steps to program.



Automated Analysis



… It is easier then you think:

32 lines of code on-top of dataframe

4 lines to automate

~ ¾ of a days work

Common Issues

The errors produced by R are not exactly intuitive

```
> #Checks the R1 file against file names in the info file
> fileName <- strsplit(fastQFileR1,'/')[[1]]
Error in strsplit(fastQFileR1, "/")[[1]] : subscript out of bounds
>
```

If you are seeing errors, then it is likely caused by the following issues:

1). You've given it the wrong file path (R can be very picky).

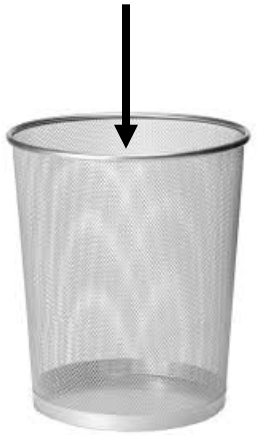
2). You have no (or very few) reads in your data (check the CDR3 translator output file sizes).

3). AutoTCR or one of its dependencies has not been installed correctly.

A Quick Note About Subsampling

- Randomly subsampling all of your samples to the **same depth** is **essential** for comparison.

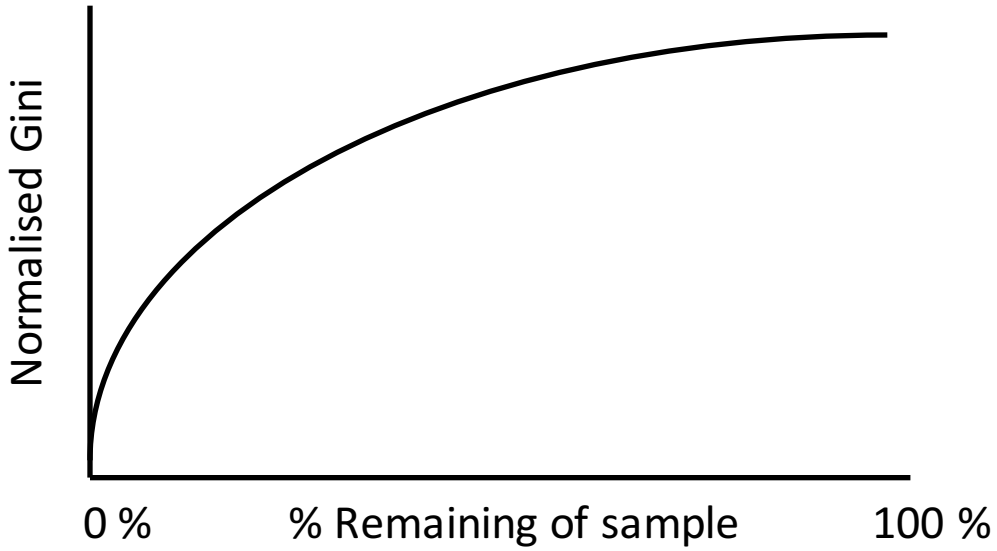
- However, **subsampling too far down** (i.e. < 10 %) is **just as bad** as not subsampling your data. You are literally **throwing away** > 90 % of your data!

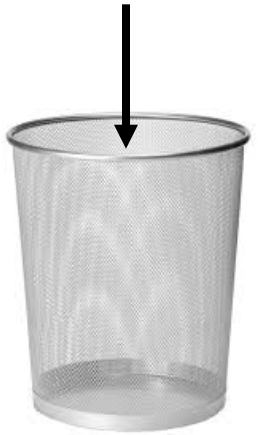


A Quick Note About Subsampling

- Randomly subsampling all of your samples to the **same depth** is **essential** for comparison.

- However, **subsampling too far down** (i.e. < 10 %) is **just as bad** as not subsampling your data. You are literally **throwing away** > 90 % of your data!

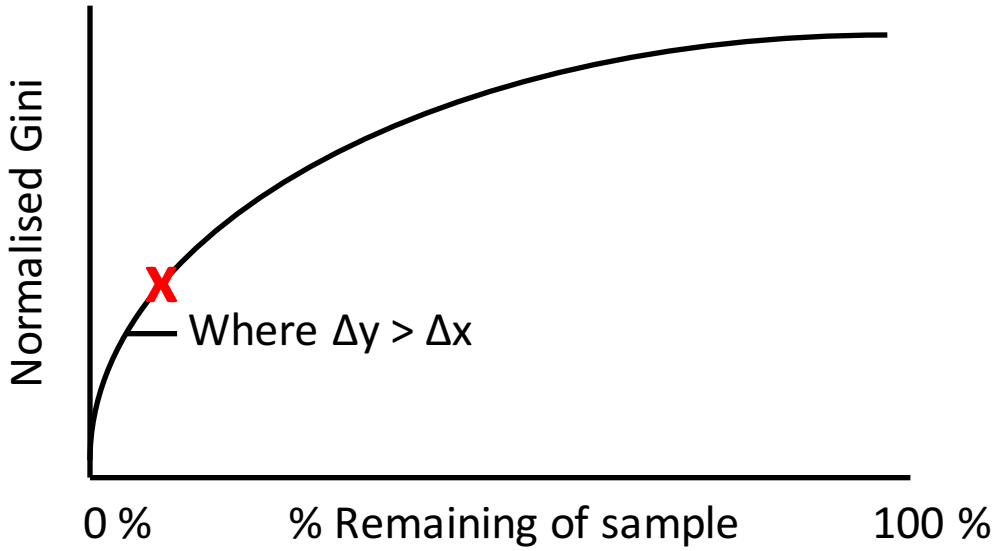


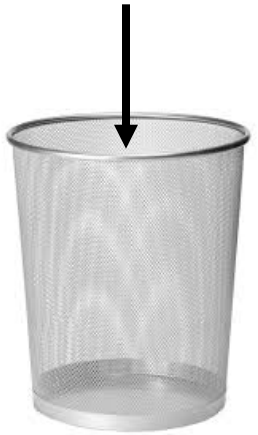


A Quick Note About Subsampling

- Randomly subsampling all of your samples to the **same depth** is **essential** for comparison.

- However, **subsampling too far down** (i.e. < 10 %) is **just as bad** as not subsampling your data. You are literally **throwing away** > 90 % of your data!





A Quick Note About Subsampling

X (position) is very variable between each sample

We calculate the value of **X** for each sample

Next we calculate a subsampling depth that includes the maximum number of samples

A Quick Note About Subsampling

X (position) is very variable between each sample

We calculate the value of **X** for each sample

Next we calculate a subsampling depth that includes the maximum number of samples

If this isn't considered, you risk artificially seperating patient populations

Control

Normal cell count

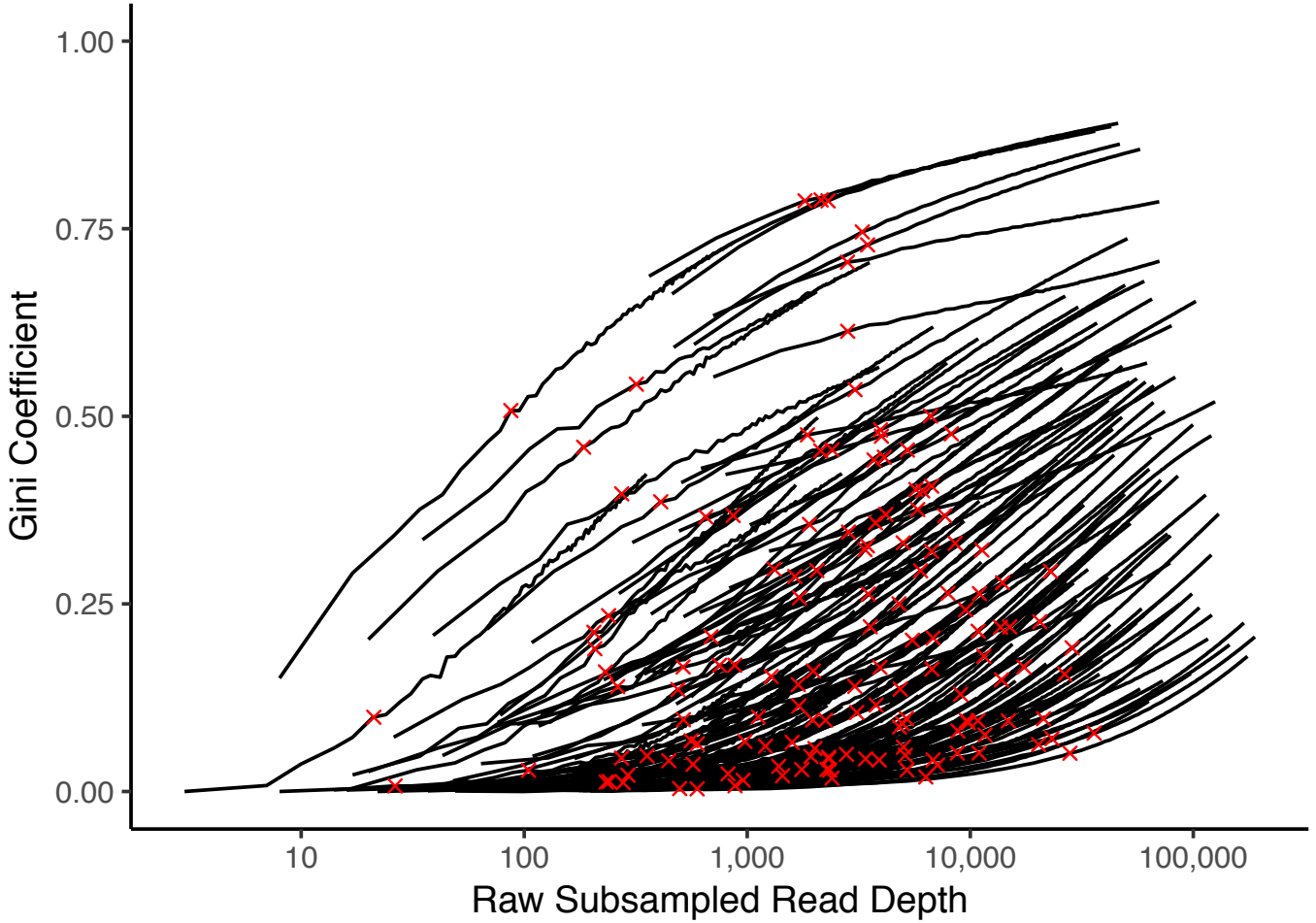
Large number of reads

Already good diversity

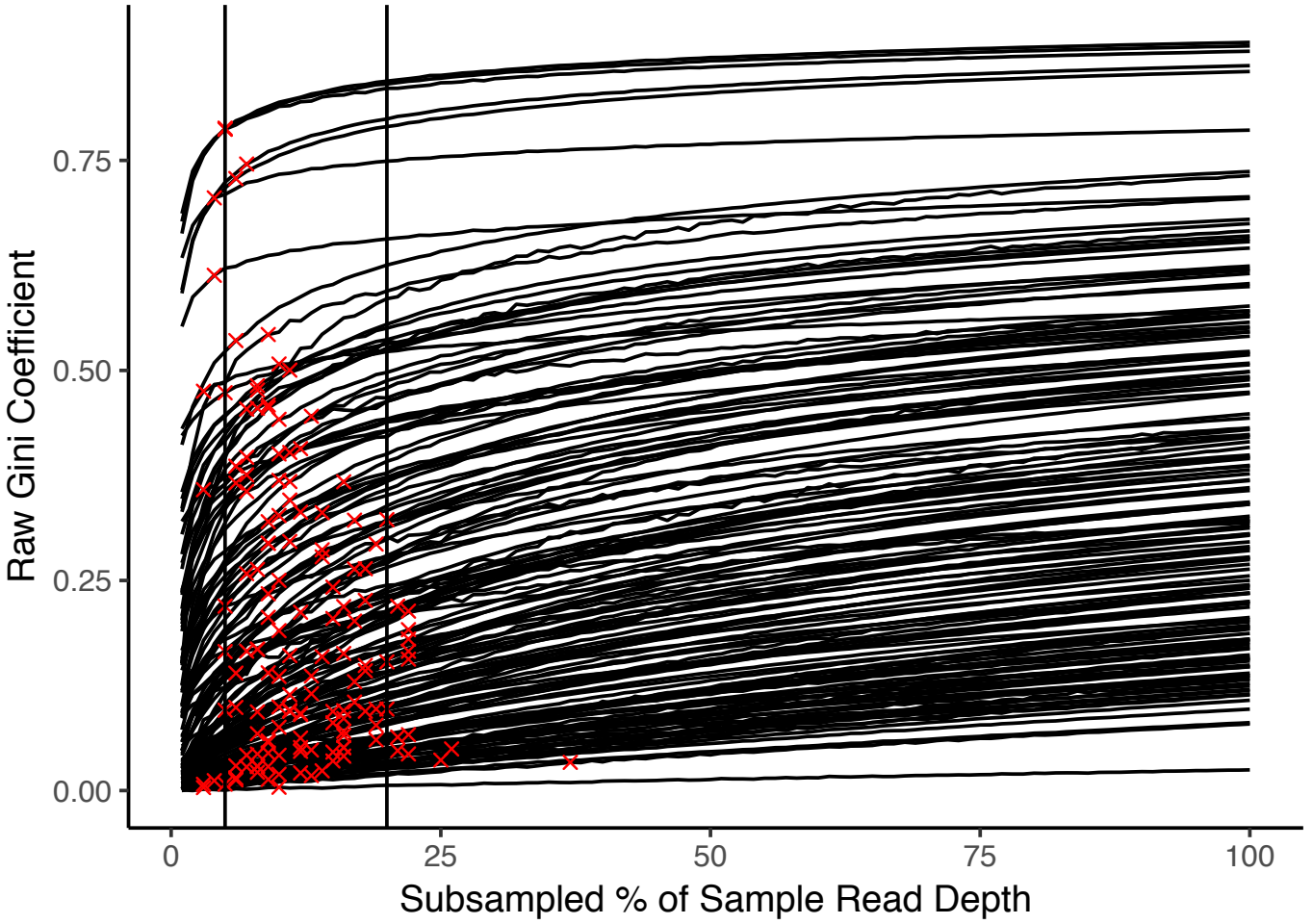
Needs sig. subsampling to be compared with…

Patient

A Quick Note About Subsampling



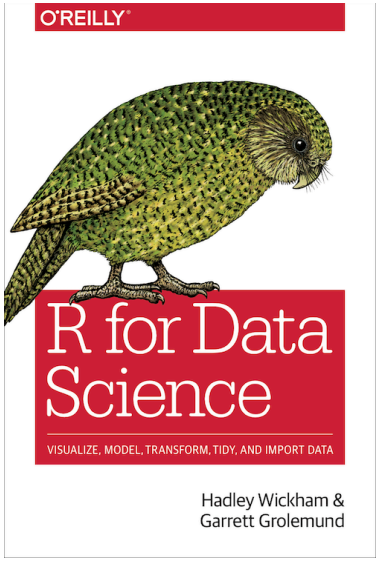
A Quick Note About Subsampling

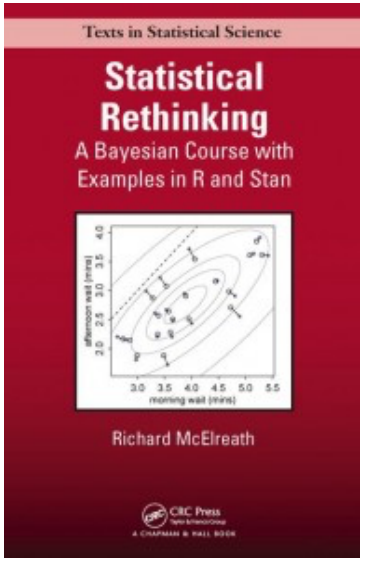


Normalising is key!

Resources

- Both of us feel strongly that **equal time** between lab and analysis is essential.
- http://r4ds.had.co.nz (Hadley Wickham)
- Statistical Rethinking (Richard McElreath)





- https://www.codecademy.com
- https://learnpythonthehardway.org
- https://www.kaggle.com
- https://www.tidyverse.org
- https://pandas.pydata.org/

(Last tip – backup everything)

Suggested Future Work
(The things we ran out of time to do)

- We worked with the data and resources available.
- Thinking of the next steps is key (lab techniques + data analysis)

Lab

- Automated library prep (Hamilton)
- B cell sequencing
- High-throughput sequencing (MiSeq -> NextSeq/HiSeq?)
- Single cell sequencing

Data Analysis

- Further streamlining of AutoTCR/Package publishing
- Interactive analysis
- Integrating diversity w/ cell counts
- Machine learning
- Mixture models

Questions?

Acknowledgements

Athina Soragia Gkazi (Lab)
Stuart Adams (Lab)
Katrine Schou-Sandgaard (Lab)
Rose Gkouleli (Lab)
Lana Mhaldien (Lab)
Theres Oakes (Lab)
Eleanor Watt (Lab)
Robin Callard (PI)
Graham Davies (PI)
Joseph Standing (Supervisor)
Nigel Klein (Supervisor)
Judith Breuer (Supervisor)
Jamie Heather (Software)
Richard Goldstein (Modelling)
Watjana Lilaonitkul (Modelling)
Theresa Attenborough (Data)
John Booth (Data)
Benny Chain (Data)