Quantifying T cell repertoire diversity

This practical uses the tcR package [4] to quantify the T cell repertoire data published by Warren *et al.* [5]. You will learn

- how to analyze complicated next generation sequencing (NGS) data on T cell repertoires
- to work with R using the tcR package
- how to study the gene-usage and diversity of a T cell repertoire
- to appreciate how difficult it is to estimate the diversity of the repertoire.
- you will become familiar with various measures for repertoire diversity and learn to read rarefaction curves.

Warren *et al.* [5] sequenced TCRβ genes from two blood draws that were taken 1 week apart from a healthy male volunteer. Each blood draw contained about 10×10^6 cells and was sequenced on several lanes of an Illumina sequencer. We analyze this data because it is publicly available, and because one can estimate the diversity of the repertoire from the incidence of TCRs in the different lanes by the Chao2 estimator [2]. The data was error-corrected by the R-TCR package [3], and the files from every lane are available in the directories `draw1` and `draw2`. Because there is less data in blood draw 2 we will start with that data set.

The manual of tcR can be found on `https://cran.r-project.org/web/packages/tcR/tcR.pdf`, and a tutorial is available on `http://imminfo.github.io/tcr/tcrvignette.html`. All documents for the practical can be found on the webpage: `http://tbb.bio.uu.nl/rdb/practicals/tcr`. Make a local directory (folder) on your computer and save the `practical.R` and `tcR_RTR_parser.R` files in that directory. In the same directory make a directory called `draw2`, and save the six `.tsv` files into that directory. Because it is convenient to use RStudio, you should open `practical.R` in the RStudio interface. You probably have to install the tcR-library: type `install.packages(tcR)` in the console, or use `Install Packages` in the `Tools` menu. You probably also have to set the working directory to the folder where you stored all the files (`Set working directory` in the `Session` menu). Then slowly proceed through the `practical.R` script by running it line-by-line (using `Control Enter`). Make sure that you understand what is happening, and make notes.

Project 1. Read the data and look at the first few lines of the data structure to see what is in the data (e.g., by typing `head(lanes[[1]])`). Next provide answers to the following biological questions:

1. How many distinct TCRβ sequences are present in each library of blood draw 2? Hint: use the `repseq.stats()` from the tcR package (which also gives the coverage of the data). Note that the number of distinct species defines the diversity of each lane (this simplest measure of diversity is also known as species "richness").
2. Are all libraries exhaustively sequenced (i.e., do you think the coverage is sufficient)?
3. How many distinct TCRβ sequences are present in blood draw 2? Why is this not the sum of the richness of all the lanes?
4. Study the distribution of the clone sizes (e.g., by typing `hist(log10(lanes[[1]]$Read.count))`). How many clones are large, how many are small? Hint: use the `top.proportion()` and `clonal.space.homeostasis()` functions of tcR. What is the difference between these two functions, and which one do you find most informative?
5. What is a Simpson diversity, and what is the Simpson diversity of the lanes and the whole blood draw2? Hint: read the definition of Simpson diversity on Wikipedia, and use the `repDiversity()` function of tcR. Note that this function knows several more diversity measures.
6. What is the overlap in TCRs between the various libraries? Hint: use `repOverlap`, and study the different measures for the overlap (e.g., `exact` and `jaccard`).
7. The 6 Illumina lanes actually came from 5 PCR libraries. Can you tell which two lanes are from the same library?

Project 2. Human TCR β chains are formed by random rearrangement of 50 V and 13 J gene segments, and all clones using a particular V-J combination can be called a family. Let's study the frequencies with which the different V and J-segments are used in the repertoire.

1. Are there any V gene segments that are more commonly used than others? Is that a consistent pattern in all libraries? Hint: use the `geneUsage()` function.
2. Which J gene segments are very common in this volunteer?
3. Investigate a couple of V-J families more closely by studying their diversity. Start with a large family. Hint: a family can be created using the `subset()` function of R.
4. Study the amino acid usage of the families you picked. For instance, save all TCRs of a given length from a large V-J family to generate a sequence logo. Hint: use `subset()` to select a certain length and `write.table()` to print al CDR3s to a file, and use WebLogo (`http://weblogo.berkeley.edu/logo.cgi`) to generate the logo. Can you predict the germline sequence underlying this family?

Project 3. Now that we know how to study the lengths of CDR3 sequences, we can repeat the analysis of Arstila *et al.* [1]. They were ahead of their time because NGS did not exist yet, and they estimated the diversity of TCR β repertoire by Sanger sequencing all CDR3s of one particular length in a given V-J family. Knowing the frequency with which this V-J combination is used (from antibody data) and knowing the fraction of sequences of that length within the family (from the relative fluorescence intensity of the band (finger) on the gel in which the TCRs were separated on length by electrophoresis), they extrapolated from this single finger to the whole repertoire. Due to the limited sequencing technology in those days, the estimate was based upon just a few sequences (about 10–20), and hence the extrapolation seems bold. We can now test how bold because we have many more sequences, and we know for all families the number of clones of any length, and from the usage of the V-J combinations we know the frequencies of all families.

1. Pick a few families and a few lengths and repeat their analysis. Try to include a finger with 10–20 TCRs.
2. How robust is their estimate?

Project 4. Can you estimate the diversity (richness) of the whole repertoire? Rarefaction curves provide an interesting graphical approach to study whether the samples were sufficiently large to estimate the total body diversity. Note that there are two issues here: we need to sample enough cells to have a representative sample of the total body repertoire, and this sample has to be sequenced deep enough such that all unique TCRs are detected. Another approach is the Chao2 diversity estimate, which uses the incidence of clones in the samples [2]. The ratio of singletons over doubletons provides a measure for the diversity because a large number of species that are present in just one sample indicates that the total diversity is much larger than the current diversity of the samples.

1. Make sure you understand what a rarefaction curve is, and make them using the `rarefaction()` function of tcR.
2. Compute the Chao2 diversity using the function provided in the R-script.
3. Since blood draw one is much larger (and has sequencing 22 lanes made from 11 PCR libraries), one could add these samples to obtain a better estimate of the diversity. So save the lanes of blood draw 2 (`lanes2 <- lanes`) and return to the beginning of the script to read the lanes from blood draw one. Check the richness and the coverage of all lanes in blood draw one, and infer which lanes came from the same library.
4. Use the lanes from both blood draws to estimate the diversity. How many TCRs do you have in both samples? Do the rarefaction curves suggest saturation? What is the Chao2 estimate?
5. Do you think you have a fair estimate of the diversity of the TCR repertoire of this volunteer?

June 28, 2016. Bram Gerritsen, Aridaman Pandit & Rob J. de Boer

References

[1] **Arstila, T. P., Casrouge, A., Baron, V., Even, J., Kanellopoulos, J., and Kourilsky, P.**, 1999. A direct estimate of the human alphabeta T cell receptor diversity. Science **286**:958–961.

[2] **Chao, A. and Bunge, J.**, 2002. Estimating the number of species in a stochastic abundance model. Biometrics **58**:531–539.

[3] **Gerritsen, B., Pandit, A., Andeweg, A. C., and de Boer, R. J.**, 2016. Rtcr: a pipeline for complete and accurate recovery of t cell repertoires from high throughput sequencing data. Bioinformatics .

[4] **Nazarov, V. I., Pogorelyy, M. V., Komech, E. A., Zvyagin, I. V., Bolotin, D. A., Shugay, M., Chudakov, D. M., Lebedev, Y. B., and Mamedov, I. Z.**, 2015. tcR: an R package for T-cell receptor repertoire data analysis. BMC Bioinformatics **16**:175.

[5] **Warren, R. L., Freeman, J. D., Zeng, T., Choe, G., Munro, S., Moore, R., Webb, J. R., and Holt, R. A.**, 2011. Exhaustive T-cell repertoire sequencing of human peripheral blood samples reveals signatures of antigen selection and a directly measured repertoire size of at least 1 million clonotypes. Genome Res. **21**:790–797.

practical.R

```
#install.packages("tcR")

source("tcR_RTCR_parser.R")
library(tcR)
library(tools)

# Read all lanes from the directory "dir"
lanes <- list()
dir <- "draw2"
filenames <- list.files(dir)
for (i in 1:length(filenames)){
  lanes[[i]] <- parse.rtcr(file.path(dir, filenames[[i]]))
}
names(lanes) <- file_path_sans_ext(filenames)

# Merge all lanes into a separate data frame
merged <- group.clonotypes(do.call(rbind, lanes), .gene.col=NA, .seq.col="CDR3.nucleotide.
    sequence")

# Look at the data
head(lanes[[1]])
str(lanes[[1]])

# How many TCRs in every lane, how many in all?  What is the Simpson diversity?
repseq.stats(lanes)
repseq.stats(merged)
repDiversity(lanes, 'inv.simp', 'read.prop')
repDiversity(merged, 'inv.simp', 'read.prop')

# What is the clone size distribution?
hist(log10(lanes[[1]]$Read.count))
top.proportion(lanes, 10)   # Fraction of ten largest clones
top.proportion(lanes, 100)
vis.top.proportions(lanes)  # Graphical representation

bins <- clonal.space.homeostasis(lanes) # Binning clonesizes
bins
vis.clonal.space(bins) # Graphical representation

# What is the overlap between the lanes?
repOverlap(lanes, 'exact', .norm = F)
vis.heatmap(intersectClonesets(lanes))
repOverlap(lanes, 'jaccard')
vis.heatmap(repOverlap(lanes, 'jaccard'))

# What is the V and J gene usage?
geneUsage(lanes, HUMAN_TRBV)
vis.gene.usage(geneUsage(merged, HUMAN_TRBV))
vis.gene.usage(geneUsage(lanes, HUMAN_TRBV), .dodge = T)
vis.gene.usage(geneUsage(lanes, HUMAN_TRBJ), .dodge = T)
vis.gene.usage(geneUsage(lanes, list(HUMAN_TRBV, HUMAN_TRBJ)))

# What is the CDR3 length distribution of one family in the merged data?
family <- subset(merged, V.gene == "TRBV20-1" & J.gene == "TRBJ2-1")
repseq.stats(family)
vis.count.len(family)
vis.count.len(family) + scale_x_continuous(breaks=seq(27,100,by=3))
finger <- subset(family, nchar(CDR3.nucleotide.sequence) == 45)
write.table(finger$CDR3.amino.acid.sequence, file = "finger.csv", row.names = F, quote = F,
    col.names = F)

# Redo the Artila analysis for this finger
repseq.stats(finger)
repseq.stats(family)
finger_frac <- repseq.stats(finger)["Sum.reads"] / repseq.stats(family)["Sum.reads"]
family_frac <- repseq.stats(family)["Sum.reads"] / repseq.stats(merged)["Sum.reads"]
arstila <- repseq.stats(finger)["Clones"] / finger_frac / family_frac
arstila/repseq.stats(merged)["Clones"]

# Make Rarefaction curves for each lane and for the merged data
rar <- rarefaction(lanes, .col = "Read.count")
vis.rarefaction(rar)
rar <- rarefaction(merged, .col = "Read.count")
vis.rarefaction(rar)
```

```
# Compute the Chao2 diversity from the incidence of each TCR in all lanes
# s = number of observed species; n = number of samples
# f1 = number of species occuring only once; f2 = species occuring twice
chao2 <- function(s, n, f1, f2){
  if(f2 == 0){
    return(s + ((n-1)/n) * f1*(f1-1)/(2*(f2+1)))
  }else{
    return(s + ((n-1)/n) * (f1^2)/(2*f2))
  }
}

shared <- shared.repertoire(lanes)
tab <- table(shared$People)
chao2(sum(tab), length(lanes), tab["1"], tab["2"])

lanes2 <- lanes    # Save your lanes

# Now read the lanes from blood draw 1, i.e., goto above, and redo what you like
# Then make a rarefaction curve and compute the Chao2 index of all data

all_lanes <- append(lanes, lanes2)    # Combine the 22 + 6 lanes
merged <- group.clonotypes(do.call(rbind, all_lanes), .gene.col=NA, .seq.col="CDR3.nucleotide.
    sequence")
repseq.stats(merged)
rar <- rarefaction(merged, .col = "Read.count")
vis.rarefaction(rar)

shared <- shared.repertoire(all_lanes)
tab <- table(shared$People)
chao2(sum(tab), length(all_lanes), tab["1"], tab["2"])
```