Extended Experimental Procedures

Subject Recruitment

Poly(A) selected RNA-Seq samples (n=38). In this analysis, we used a subset of Puerto Rican Islanders recruited as part of the on-going Genes-environments & Admixture in Latino Americans study (GALA II).[1–4] We classified asthma by physician diagnosis and the presence of at least two symptoms (wheezing, coughing, or shortness of breath) during 2 years prior to the enrollment. All study subjects had no history of smoking or recent (within 4 weeks of recruitment) nasal steroid use. The study was approved by local institutional review boards, and written assent/consent was received from all subjects and, if applicable, parents of subjects under the age of legal consent.

Ribo-Zero RNA-Seq samples (n=49). Via community-based advertising, we recruited adults aged 18-70 years to participate in a study, in which they underwent research bronchoscopy. The study was approved by the University of California at San Francisco Committee on Human Research. Written informed consent was obtained from all subjects, and all studies were performed in accordance with the principles expressed in the Declaration of Helsinki.

Sample Collection

Poly(A) selected RNA-Seq samples (n=38). Methods for nasal epithelial cell collection and processing are described in Poole et al.[4] Briefly, nasal epithelial cells were collected from behind the inferior turbinate with a cytology brush using a nasal illuminator. The

collected brush was submerged in a mixture of RLT Plus lysis buffer and beta-mercaptoethanol, and frozen at -80 C until extraction was performed with a Qiagen Allprep RNA/DNA extraction kit (Qiagen, Valencia, CA). We collected 10ml of whole blood using PAXgene RNA blood tubes (PreAnalytiX, Valencia, CA) and isolated RNA using PAXgene RNA blood extraction kits, according to the manufacturers' protocol. Portions of the nasal airway epithelial whole transcriptome data were published in a previous manuscript.[4]

Ribo-Zero RNA-Seq samples (n=49). During bronchoscopy airway epithelial brushings, samples were obtained from 3rd-4th generation bronchi. RNA was extracted from the epithelial brushing samples using the Qiagen RNeasy mini-kit (Qiagen, Valencia, CA), according to manufacturer's protocol.

Whole Transcriptome Sequencing

Poly(A) selected RNA-Seq samples (n=38). We constructed poly-A RNA-seq libraries using 500 ng of blood and nasal airway epithelial total RNA from 9 atopic asthmatics and 10 non-atopic controls. Libraries were constructed and barcoded with the Illumina TruSeq RNA Sample Preparation v2 protocol. Barcoded nasal airway RNA-seq libraries from each of the 19 subjects were pooled and sequenced as 2 x 100bp paired-end reads across two flow cells of an Illumina HiSeq 2000. Barcoded blood RNA-seq libraries from each of the 19 subjects were pooled and sequenced as 2 x 100bp paired end reads across 4 lanes of an Illumina Hiseq 2000 flow cell.

Ribo-Zero RNA-Seq samples (n=49). We used 100ng of isolated RNA from a total of 61 samples to construct ribo-depleted RNA-seq libraries using the TruSeq Stranded Total RNA with Ribo-Zero Human/Mouse/Rat library preparation kit, per manufacturer's protocol. Barcoded bronchial epithelial RNA-seq libraries were multiplexed and sequenced as 2 x 100bp paired end reads on an Illumina HiSeq 2500. On average, 37 million reads were generated per sample. We excluded 12 samples from further analyses due to high ribosomal RNA read counts (library preparation failure), leaving a total of 49 samples suitable for further analyses

Workflow to categorize the mapped reads

Map reads onto human genome and transcriptome

We mapped reads onto the human transcriptome (Ensembl GRCh37) and genome reference (Ensembl hg19) using tophat2 (v 2.0.13) with the default parameters. Tophat2 was supplied with a set of known transcripts (as a GTF formatted file, Ensembl GRCh37) using –G option. The mapped reads of each sample are stored in a binary format (.bam).

Categorize mapped reads into genomic categories

ROP categorizes the reads into genomic categories based on the compatibility of each read from the pair with the features defined by Ensembl (GRCh37) gene annotations. First, we determined CDS, UTR3, UTR5 coordinates. We downloaded annotations for CDS, UTR3, UTR5 from UCSC Genome Browser (http://genome.ucsc.edu/cgi-bin/hgTables) in BED (browser extensible data) format. Next, we used gene annotations (a GTF formatted

file, Ensembl GRCh37) to determine intron coordinates and inter-genic regions. We defined two types of inter-genic regions: '(proximate) inter-genic' region (1Kb from the gene boundaries) and 'deep inter-genic' (beyond a proximity of 1Kb from the gene boundaries).

Next, we checked the compatibility of the mapped reads with the defined genomic features, as follows:

a. Read mapped to multiple locations on the reference genome is categorized as a multi-mapped read.
b. Read fully contained within the CDS, intron, UTR3, or UTR5 boundaries of a least one transcript is classified as a CDS, intronic, UTR3, or UTR5, respectively.
c. Read simultaneously overlapping UTR3 and UTR5 regions is classified as a UTR read.
d. Read spanning exon-exon boundary is defined as a junction read.
e. Read mapped outside of gene boundaries and within a proximity of 1Kb is defined as a (proximal) inter-genic read.
f. Read mapped outside of gene boundaries and beyond the proximity of 1Kb is defined as a deep inter-genic read.
g. Read mapped to mitochondrial DNA (MT tag in hg19) is classified as a mitochondrial read.

h. Reads from a pair mapped to different chromosomes are classified as a fusion reads

Scripts to categorize mapped reads into genomic categories are available here:

https://sergheimangul.wordpress.com/gprofile/

Categorize mapped reads overlapping repeat instances

Mapped reads were categorized based on the overlap with the repeat instances defined by RepeatMasker annotation (Repeatmasker v3.3, Repeat Library 20120124). RepeatMasker masks the repeats using the RepBase library: (http://www.girinst.org/repbase/update/index.html), which contains prototypic sequences representing repetitive DNA from different eukaryotic species. We use GTF files generated from the RepeatMasker annotations by Jin, Ying, et al.[1] and downloaded from:

http://labshare.cshl.edu/shares/mhammelllab/www-data/TEToolkit/TE_GTF/hg19_rmsk_TE.gtf.gz

Following Melé, Marta, et al.[2], repeat elements overlapping CDS regions are excluded from the analysis. We filtered out 6,873 repeat elements overlapping CDS regions. Prepared repeat annotations (bed formatted file) are available here:

https://sergheimangul.wordpress.com/rop/repeats/

The prepared repeat annotations contain 8 Classes and 43 Families. Number of elements per family and class represented below:

classID	N
DNA	**458223**
LINE	1478382
LTR	707384
RC	2226
SVA	3582
RNA	717
Satellite	8950
SINE	1765403

Table 1. Number of repeat elements per class. Repeat instances are defined by RepeatMasker (RepeatMasker v3.3, Repeat Library 20120124) based on RepBase library. RepBase library contains prototypic sequences representing repetitive DNA from different eukaryotic species.

familyID	n

acro	44
Alu	1173282
centr	2272
CR1	60577
Deu	1262
DNA	4609
Dong-R4	554
ERV	579
ERV1	172612
ERVK	10446
ERVL	159606
ERVL-MaLR	343266
Gypsy	18553
hAT	15418
hAT-Blackjack	19578
hAT-Charlie	251618
hAT-Tip100	30204
Helitron	2226
L1	937636
L2	461296
LTR	2322
Merlin	55

MIR	589496
MuDR	1978
Penelope	51
PiggyBac	2352
RNA	717
RTE	17617
RTE-BovB	651
Satellite	6247
SINE	1363
SVA_A	257
SVA_B	465
SVA_C	279
SVA_D	1358
SVA_E	232
SVA_F	991
TcMar	5354
TcMar-Mariner	16253
TcMar-Tc2	8098
TcMar-Tigger	102706
telo	387

Table 2. Number of repeat elements per family. Repeat instances are defined by RepeatMasker (RepeatMasker v3.3, Repeat Library 20120124) based on RepBase library.

We determined the coordinates of repeat elements (*class_id* and *family_id* attributes from the GTF file) from the repeat annotations. Next, we checked the compatibility of the mapped reads with the repeat instances. We disregarded the pairing information for the unmapped reads and count each end as a separate read. Reads entirely mapped to the corresponding repeat instance are counted. Scripts to categorize mapped reads based on the overlap with the repeat instances are available here: https://sergheimangul.wordpress.com/rprofile/ .

***Categorize mapped reads overlapping* B cell receptor (BCR) and T cell receptor (TCR) loci**

We used the gene annotations (Ensembl GRCh37) to extract antibody genes. We extracted gene annotations of the 'constant' (labeled as IG_C_gene, Ensembl GRCh37), 'variable' (labeled as IG_V_gene, Ensembl GRCh37), 'diversity' (labeled as IG_D_gene, Ensembl GRCh37), and 'joining' genes (labeled as IG_J_gene, Ensembl GRCh37) of BCR and TCR loci. We excluded the BCR and TCR pseudogenes (labeled as IG_C_pseudogene, IG_V_pseudogene, IG_D_pseudogene, IG_J_pseudogene, TR_C_pseudogene, TR_V_pseudogene, TR_D_pseudogene, and TR_J_pseudogene). In addition, we excluded the patch contigs *HG1592_PATCH and HG7_PATCH,* as they are not part of the Ensembl hg19 reference, and reads are not mapped on the patch contigs by high throughput

aligners. After following the filtering steps described above, we extracted a total of 386 immune genes: 207 BCR genes and 179 TCR genes. The gene annotations for antibody genes (GTF formatted file) are available here:

https://sergheimangul.wordpress.com/antibodies/

The number of VDJ genes per locus is reported in the Table 3.

	C domain	V domain	D domain	J domain
***IGH* locus**	8	55	38	6
***IGK* locus**	1	46	-	5
***IGL* locus**	4	37	-	7
TCRA locus	1	46	-	57
TCRB locus	1	39	0	8
TRG locus	2	9	-	5
TRD locus	1	3	11	4

Table 3. The number of VDJ genes for each antibody chains. Antibody genes were extracted from the gene annotations (Ensembl GRCh37).

The list of the genes encoding the C region of the BCR and TCR chains is presented in Table 4.

Name of the chain	Genes encoding for the C region of the chain
IG@ locus	
α heavy IG chain	IGHA1, IGHA2
δ heavy IG chain	IGHD
γ heavy IG chain	IGHG1, IGHG2, IGHG3, IGHG4
ε heavy IG chain	IGHE
μ heavy IG chain	IGHM
κ light IG chain	IGKC
λ light IG chain	IGLC1, IGLC2, IGLC3, IGLC7
TCR@ locus	
α TCR chain	TRAC
B TCR chain	TRBC2
γ TCR chain	TRGC1, TRGC2
δ TCR chain	TRDC

Table 4. List of the genes encoding the C region of the BCR and TCR chains. Genes were extracted from the gene annotations (Ensembl GRCh37).

The number of reads mapping to each C-V-D-J genes was obtained by counting the number of sequencing reads that align, with high confidence, to each of the genes (HTSeq v0.6.1)[3]. Script "htseq-count" is supplied with the gene annotations for antibody genes (genes_ Ensembl_GRCh37_BCR_TCR.gtf) and a bam file. The bam file contains reads

mapped to the human genome and transcriptome using tophat2 (See Section ***"Map reads onto human genome and transcriptome"*** for details). The script generates individual gene counts by examining the read compatibility with BCR and TCR genes. We chose a conservative setting (--mode=intersection-strict) to handle reads overlapping more than one feature. Thus, a read overlapping several genes simultaneously is marked as a read with no feature and is excluded from the consideration.

Workflow for categorizing the unmapped reads

We first converted the unmapped reads saved by tophat2 from a BAM file into a FASTQ file (using bamtools). The FASTQ file of unmapped contain full read pairs (both ends of a read pair were unmapped) and discordant read pairs (one read end was mapped while the other end was unmapped). We disregarded the pairing information of the unmapped reads and categorize unmapped reads using the following steps:

A. Quality Control

Low quality reads, defined as reads that have quality lower than 30 in at least 75% of their base pairs, were identified by FASTQC. Low complexity reads, defined as reads with sequences of consecutive repetitive nucleotides, are identified by SEQCLEAN. As a part of the quality control, we also excluded unmapped reads mapped onto the rRNA repeat sequence (HSU13369 Human ribosomal DNA complete repeating unit) (BLAST+ 2.2.30). We prepared the index from rRNA repeat sequence using makeblastdb and makembindex from BLAST+. We used the following command for makeblastdb:

- makeblastdb -parse_seqids -dbtype nucl -in <fasta file>.

We used the following command for makembindex:

- makembindex -input <fasta file> -output <index> -iformat blastdb

B. Mapping unmapped reads onto the human references.

We remapped the unmapped reads to the human reference sequences using Megablast (BLAST+ 2.2.30). We used the following references to map the reads onto:

- Reference transcriptome (known transcripts), Ensembl GRCh37
- Reference genome, hg19 Ensembl

We prepared the index from each reference sequence using makeblastdb and makembindex. We mapped the reads separately onto each reference in the order listed above. Reads mapped to the reference genome and transcriptome were merged into a 'lost human reads' category. The following options were used to map the reads using Megablast: for each reference: task = megablast, use_index = true, perc_identity = 90, outfmt = 6, max_target_seqs =1, e-value = $1e^{-05}$. Reads not entirely mapped are discarded (e.g. 'alignment length' < read length).

C. Mapping unmapped reads onto the repeat sequences

We filtered out the reads that failed QC and lost human reads. The remaining reads were mapped to the reference repeat sequences. The reference repeat sequences were downloaded from Repbase v20.07 (http://www.girinst.org/repbase/). Human repeat elements (humrep.ref and humsub.ref) were merged into a single reference. We prepared the index from the merged repeat reference using makeblastdb and makembindex from BLAST+. In total, we obtained sequences for 1,117 repeat elements. The following options were used to map the reads using the Megablast: task = megablast, use_index = true, perc_identity = 90, outfmt = 6, max_target_seqs = 1, e-value = $1e^{-05}$. Blast hits with alignment length shorter than 80% of the read length were discarded (corresponding to 80bp of the 100bp read).

The repeat elements from humrep.ref and humsub.ref were classified into families and classes using RepeatMasker annotations (hg19_rmsk_TE_prepared_noCDS.bed). Repetitive reads identified from the unmapped reads were confirmed by directly applying Repeatmasker[4].

***D. Workflow to* detect 'non-co-linear' reads (trans-splicing, gene fusions, and circRNAs)**

We have developed a custom pipeline to identify reads spliced distantly on the same chromosome supporting trans-splicing events; reads spliced across different chromosomes supporting gene fusion events; and reads spliced in a head-to-tail configuration supporting circRNAs: ncSplice v 1.0 which is available at https://github.com/Frenzchen/ncSplice. An overview of this pipeline is described in Figure 1. First, we filtered out the reads identified in steps (A)-(C) and used the remaining

reads to identify circRNA, intra-chromosomal (trans-splicing), and inter-chromosomal (gene fusion) reads. We extracted the terminal 20 bp from both ends of each unmapped read. Next, we remapped anchor pairs independently from each other to the canonical chromosomes of the human genome (human hg19, GRCh37) using bowtie2 (v2.1.0). The Bowtie2 command for 1st mapping is:

- bowtie2 –x <bt2-idx> -q –U anchors.fastq –S mapping.sam

Anchor pairs that map within 100 kb from each other, on the same chromosome and strand, and in a head-to-tail configuration, were considered circRNA candidates reads. Anchor pairs that map on the same chromosome, but are more than 1 Mb apart from each other, were considered intra-chromosomal fusion candidates. Anchor pairs that map on different chromosomes were defined as inter-chromosomal fusion candidates. Anchor pairs for fusion candidates were allowed to fall on different strands. Next, we extended the anchor alignments for all candidate anchor pairs. We allowed maximum of two mismatches during the extension procedure. We discarded anchor pairs for which the breakpoint was detected with an uncertainty of more than 8 bp. For all candidate breakpoints, we extracted 100 bp of flanking sequence upstream and downstream. Next, we used these to build separate junction indices for circRNAs, intra-chromosomal, and inter-chromosomal fusions. All unmapped reads were remapped with bowtie2 on theses indices to capture reads that span the junction between 8 bp and 20 bp. The Bowtie2 commands for 2nd mapping is:

- bowtie2-build -f <reference_in> <bt2_index_base>
- bowtie2 –x <bt2-idx> -q –U unmapped.fastq –no-unal –S remapping.sam

The results from two mapping rounds were then joined.

Post-detection filtering steps

Z-score: The z-score is a statistical measure, which indicates how many standard deviations an element deviates from the sample mean. In many samples, we observed loci that were highly enriched in the reads. These loci seem to contain repetitive sequences and, therefore, reads mapping to these regions are likely to be artifacts. We calculated z-scores for all read counts/event for each sample and detection type (circRNA, intra- or inter-chromosomal fusion). Events with the z-score higher than 1.96 were discarded.

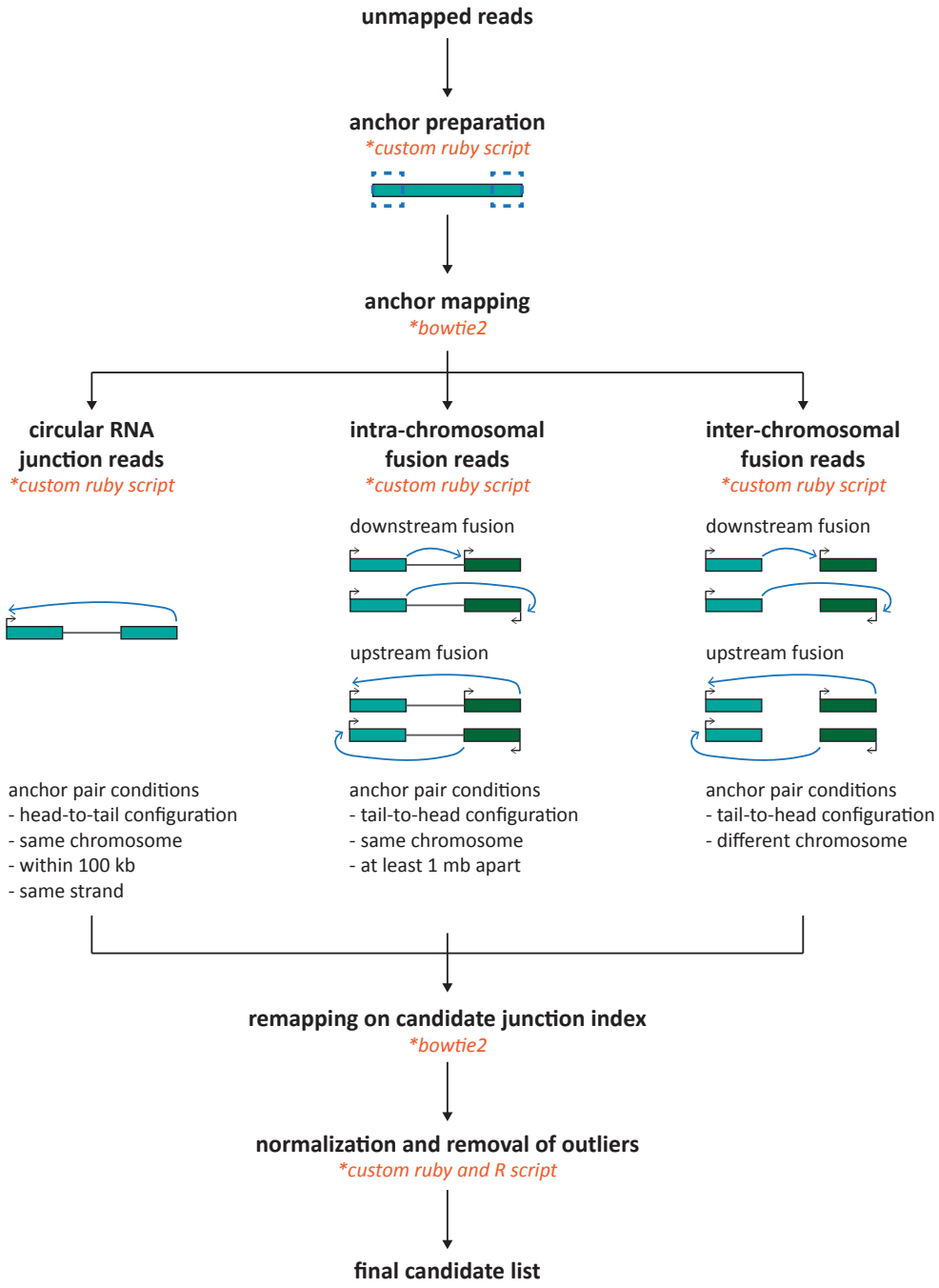

Figure 1. Overview of the ncSplice pipeline. ncSplice detects reads spliced distantly on the same chromosome supporting trans-splicing events; reads spliced across different

chromosomes supporting gene fusion events; and reads spliced in a head-to-tail configuration supporting circRNAs.

E. Mapping unmapped reads onto the V(D)J genes of antibody loci

Gene segments of B cell receptors (BCR) and T cell receptors (TCR) were imported from IMGT (International ImMunoGeneTics information system): (http://www.imgt.org/vquest/refseqh.html#V-D-J-C-sets).

IMGT database contains:

- Variable (V) gene segments
- Diversity (D) gene segments
- Joining (J) gene segments

Unmapped reads categorized by step (A)-(D) were filtered out. We used IgBLAST (v. 1.4.0) with stringent e-value threshold (e-value < 10^{-20}) to map the remaining high-quality unmapped reads onto the V(D)J regions of the of the BCR and TCR loci. Reference files with BCR and TCR VDJ gene segments are distributed with ROP protocol and available here:

https://sergheimangul.wordpress.com/vdj/

The complete list of the references is presented in Table 5.

Name of the reference file	Description of the gene
BCR heavy chain	
IGHV.fa	V genes of BCR heavy chain
IGHD.fa	D genes of BCR heavy chain
IGHJ.fa	J genes of BCR heavy chain
BCR light chains	
IGLV.fa	V genes of BCR lambda chain
IGLJ.fa	J genes of BCR lambda chain
IGKV.fa	V genes of BCR kappa chain
IGKJ.fa	J genes of BCR kappa chain
TCR chains	
TCRAV.fa	V genes of TCR alpha chain
TCRAJ.fa	J genes of TCR alpha chain
TCRBV.fa	V genes of TCR beta chain
TCRBD.fa	D genes of TCR beta chain
TCRBJ.fa	J genes of TCR beta chain
TCRGV.fa	V genes of TCR gamma chain
TCRGJ.fa	J genes of TCR gamma chain
TCRDV.fa	V genes of TCR delta chain
TCRDD.fa	D genes of TCR delta chain
TCRDJ.fa	J genes of TCR delta chain

Table 5. List of the references files prepare for V-D-J from BCR and TCR loci.

We prepared the index from each reference sequence using makeblastdb and makembindex from BLAST+. The following options were used to map the reads using IgBLAST: -germline_db_V; germline_db_D; -germline_db_J; -organism=human; -outfmt = 7; –evalue = 1e-20.

The number of genes and gene alleles per antibody locus is presented in Table 6.

	V domain	D domain	J domain
***IGH* locus**	**136**(370)	**27**(34)	**9**(16)
***IGK* locus**	**100**(124)	-	**5**(9)
***IGL* locus**	**70**(111)	-	**7**(10)
TCRA locus	**54**(112)	-	**61**(68)
TCRB locus	**77**(160)	**2**(3)	**14**(16)
TRG locus	**14**(26)	-	**5**(6)
TRD locus	**8**(22)	**0**(0)	**1**(4)

Table 6. The number of V-D-J genes and gene alleles per antibody locus. Number of genes is presented in bold and number of gene alleles is presented in parenthesis. Gene and gene alleles of B cell receptors (BCR/IG) and T cell receptors (TCR) were imported from IMGT.

We assessed combinatorial diversity of the antibody repertoire by looking at the recombinations of the VJ gene segments of BCR and TCR loci. We extracted the reads spanning the V-J gene boundaries.

F. Identification of microbial reads

Unmapped reads mapping in step (A -(E) were filtered out. The remaining reads were high-quality non-human reads used to profile the taxonomic composition of the microbial communities. We used MetaPhlAn2 (Metagenomic Phylogenetic Analysis, v 2.0) to assign reads on microbial genes and to get taxonomic profile. The database of the microbial marker genes is provided by MetaPhlAn. We run MetaPhlAn in two stages as follow: the first stage identifies the candidate microbial reads (i.e. reads hitting a marker), while the second stage profiles metagenomes in terms of relative abundances – the commands used are as follow:

- metaphlan.py <fastq> <map> --input_type multifastq --bowtie2db bowtie2db/mpa -t reads_map --nproc 8 --bowtie2out
- metaphlan.py --input_type blastout <bowtie2out.txt> -t rel_ab <tsv>

The output of the first stage is a file with the list of candidate microbial reads with the microbial taxa assigned (.map file). The second stage outputs the taxonomic profile (taxa detected and its relative abundance, in tab separated format (.tsv file). We used taxa detected from stage 2 to extract the reads associated with it in stage 1.

In addition to the tool which use the curated database of taxa-specific genes, we mapped the reads onto the entire reference genomes of microbial organisms. We used Megablast (BLAST+ 2.2.30) to align reads onto the collection of bacterial, viral, and eukaryotic pathogens reference genomes. Bacterial and viral genomes were downloaded from NCBI ftp://ftp.ncbi.nih.gov/ on February 1, 2015. Genomes of eukaryotic pathogens were downloaded from EuPathDB database which is available at: http://eupathdb.org/eupathdb/.

The following parameters were used for the megablast alignment: e-value = 10^{-5}, perc_identity = 90. The Megablast hits shorter than 80% of the input read sequence were removed (corresponding to 80bp of the 100bp read).

Comparing diversity across groups

First, we sub-sampled unmapped reads to number of reads corresponding to a sample with smallest number of unmapped reads. Diversity within a sample was assessed using the richness and alpha diversity indices. Richness was defined as a total number of distinct ***events*** in a sample. We used Shannon Index(SI) incorporating richness and evenness components to compute alpha diversity, which is calculated as follows:

$$\mathrm{SI} = -\sum(p \times \log_2(p))$$

We used beta diversity (Sørensen–Dice index) to measure compositional similarities between the samples in terms of gain or loss of the events. We calculated the beta diversity for each combination of the samples, and we produced a matrix of all pair-wise sample dissimilarities. The Sørensen–Dice beta diversity index is measured taxonomically

as $1 - \frac{2J}{A+B}$, where J is the number of shared events, while A and B are the total number of events for each sample, respectively.

Percentage of unmapped reads calculation

We calculated the percentage of unmapped reads using the following formula:

$$P_{unmapped} = \frac{(N_{ud} + (N_{uc} \times 2))}{(N_{total} \times 2)}$$

where,

N_{ud} – number of discordant unmapped reads (one end is mapped, while the other end is unmapped);

N_{uc} – number of unmapped read pairs (both ends are unmapped);

N_{total} – total number of read pairs (fragments).

Complexity analysis using Capture Recapture Model

Given a sequencing experiment, the Read Origin Protocol (ROP) attempts to classify every sequenced read in the experiment to an "origin" class. These origins can be considered to be features of interest, e.g. exons, retroviral, immune, or bacterial. Since every read is assigned to only one class we can consider the reads assigned to a specific class to be a random sample from the population of possibilities within that class. This leads us to consider statistical models for population sampling, so called "capture-recapture"

models.[5]

Using capture-recapture models allows us to make statistical inferences on several quantities of interest. Of primary interest is the total number of possibilities in the feature. We shall refer to this as the feature size but is commonly known in the statistics literature as the species richness.[5,6] We also consider the number of identified possibilities within a feature as a function of the number of reads, that we shall call the complexity of the feature, in line with the notation of Daley and Smith.[7] The rate of change in the complexity curve is proportional to the probability the next read is a previously unobserved class.[8] This quantity is commonly known in the statistics literature as the mathematical coverage[9], but to avoid confusion with sequencing coverage we call this the discovery probability[10] and one minus the discovery probability will be called the saturation of the feature.

Statistical Model

Suppose we sequence N reads from an experiment. There are C feature classes, represented in the sequencing library with proportions $\pi_1, \dots, \pi_C$. Feature may overlap, so it's not necessary that the proportions sum to one. The features are all known and defined beforehand. This is in contrast to the number of classes within each feature.

Within each feature c, there are a fixed but unknown number of classes, Sc represented in the experiment. Within the feature, these are represented with relative proportions

$$p_1, \dots, p_{S_C}, \sum_{i=1}^{S_C} p_i = 1$$

If we are interested in the relative proportions within the experiment, we multiply the

relative proportion within the feature by the relative abundance of the feature within the experiment.

The problem is that we only have information on the classes that were sequenced in the experiment. We observed $D_C \leq S_C$ classes with observed frequencies x_i = # reads from class i with $\sum_{i=1}^{S_C} x_i = N_C$ and $\sum_{c=1}^{C} N_c = N$.

The problem of estimating the complexity is to estimate the number of expected distinct classes observe as a function of reads sequenced. We use the non-parametric empirical Bayes approach of Daley and Smith[7] to estimate the feature complexity curve. The limit of the feature complexity curve can be regarded as an estimate of the feature size.[11]

The discovery probability of the observed experiment is the sum of the relative proportions of the unobserved classes,

$$\sum_{\boldsymbol{i=1}}^{\boldsymbol{S_c}} \boldsymbol{p_i 1(x_i = 0)}.$$

The non-parametric empirical Bayes estimator for this quantity is given by the Good Turing formula, $(\sum_{i=1}^{S_C} \frac{1(x_i=1)}{N_C})$.

Read Complexity Analysis

We first examine the read complexity determined by the mapped start position of the first end in the read pair. We observe little difference between the two libraries for the single end complexity (Figure 2). We observe only an average of 20% and 29% of the mappable reads at the sequenced read depth. We estimate that on average all libraries are 58% saturated, that is we observed 58% of the abundance. This is natural since one would naturally sequence the most abundant reads first.

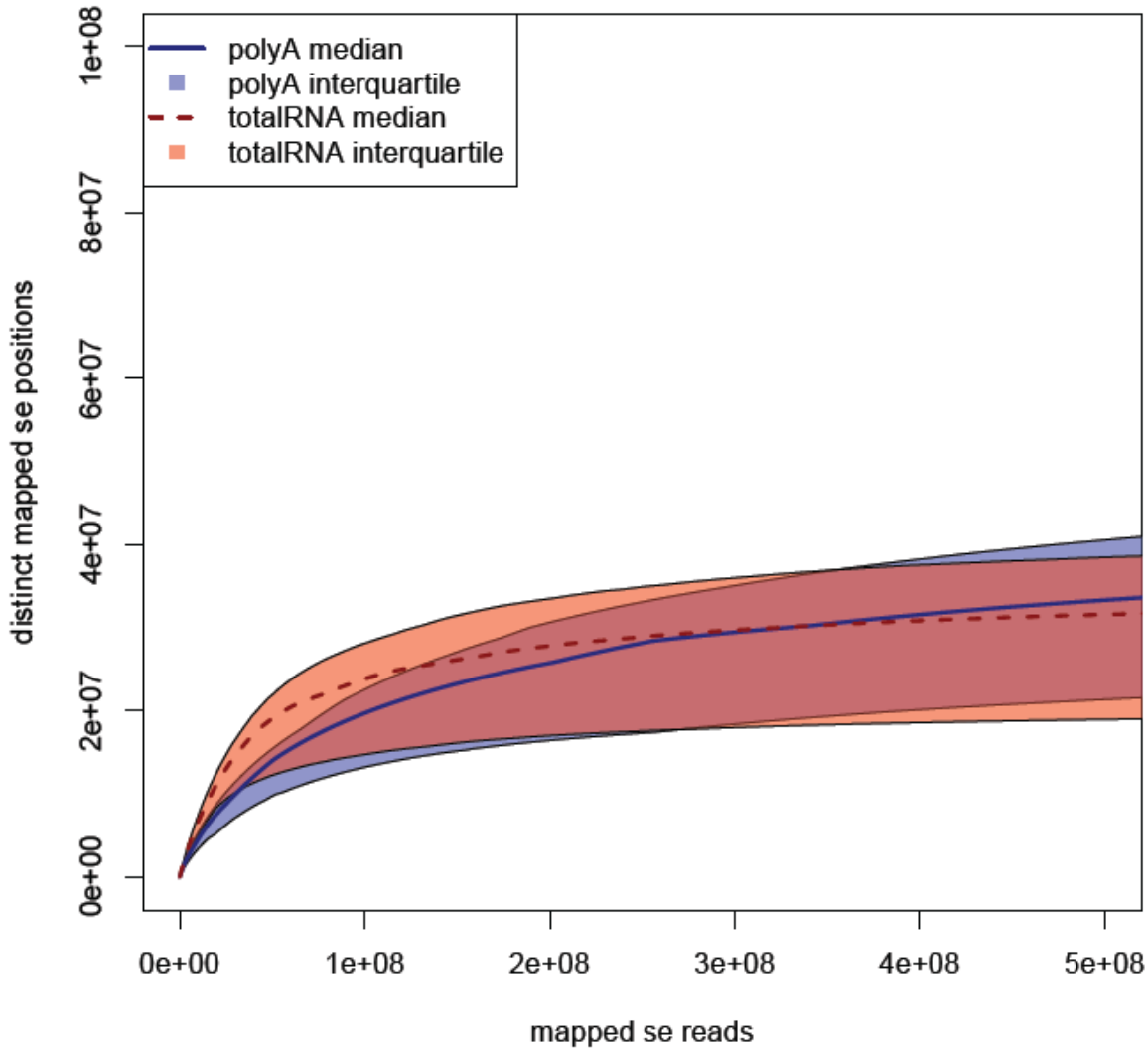

Figure 2. Single end read complexity medians and interquartile ranges across the two library preparations.

Annotated Feature Complexity Analysis

The mapped reads can be assigned to features within the genome. These include exons, introns, coding sequences (CDS), and untranslated regions (UTR). In this section we shall investigate the complexity of these features and can be interpreted as estimating the transcriptional diversity within these libraries.

As expected, more exons, CDSs, and UTRs were observed per sequenced fragment for the polyA libraries than for the totalRNA libraries. Yet all libraries are very saturated. Most of the abundant classes within these features have already been observed and the unobserved features are extremely rare. This is in line with the common practice of sequencing a few tens of millions of reads for inferring differential expression.

To compare the saturation across libraries, we extrapolated the saturation to a common value. The saturation is asymptotically normal[13] and the sequencing depth is sufficiently high that we can use a standard t-test to investigate differences. When all the features for all libraries are extrapolated out to 100 million observations the polyA libraries are more saturated (exons: p = 3.764E-16; CDS: p = 1.036E-14; UTR: p = 5.183E-14; more significant differences were observed at lower depths, indicating that the differences are not artifacts of the sampling depth).

Despite the large saturation for all features across libraries, a multitude of unobserved classes remain (Table 7). This means that most of the unobserved classes are exceedingly rare. For example, we estimate that on average there are 41,990 unobserved exons in the polyA libraries. There is an average remaining abundance of $1 - 0.9988 = 0.0012$, implying that the average abundance of the unobserved exons is $\frac{0.0012}{41990} = 2.86\,E - 8$. Since, on average, a read has $2 \cdot 0.176 = 0.352$ probability of overlapping an exon, the average abundance of the unobserved exons is 1E-8 and the total abundance, 0.00042, gives the marginal probability that the next sequenced read is a new exon. For the

totalRNA libraries, the average abundance of the unobserved exons is 3.2E-8. Similarly, we calculated the average abundance of the unobserved CDS for polyA and totalRNA libraries as 1.84E-8 and 7.78E-8, respectively, and for UTRs it was 1.1E-8 and 6.48E-8.

Finally, we examined differences of diversity between case and controls for a fixed tissue type and library type. This is quite anticlimactic, as we found little differences between cases and controls for extrapolated saturation and feature diversity. This indicates that there are little differences in transcriptome diversity between the two groups of case and controls or that the differences are so small that a much larger cohort is required to accurately infer the disparity.

Feature	Mean hits		Mean observed		Mean saturation		Mean estimated total	
	polyA	totalRNA	polyA	totalRNA	polyA	totalRNA	polyA	totalRNA
Exons	10310521		110553		0.9969		145950	
	17713362	5745436	115507	107498	0.9988	0.9956	157497	138829
CDS	4791394		105820		0.984		131521	
	8804113	2316884	116068	99500	0.9977	0.9756	144062	123788
UTR	4359596		33165		0.9948		43136	
	8035082	2093047	37448	30524	0.99913	0.99209	49849	38997

Table 7. Mean number of observations, distinct observed classes, observed saturation, and estimated total number of classes for exons, CDS, and UTR Features.

List of software tools used:

Tophat - http://ccb.jhu.edu/software/tophat/index.shtml

Bowtie2 - http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

Samtools - http://www.htslib.org/

Bamtools - https://github.com/pezmaster31/bamtools

FASTX-Toolkit - http://hannonlab.cshl.edu/fastx_toolkit/

SEQLEAN - http://sourceforge.net/projects/seqclean/files/

BLAST+ - ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

RepeatMasker - http://www.repeatmasker.org/

IgBlast - http://www.ncbi.nlm.nih.gov/igblast/

ncSplice - https://github.com/Frenzchen/ncSplice

MetaPhlAn - http://huttenhower.sph.harvard.edu/metaphlan

HTSeq - http://www-huber.embl.de/users/anders/HTSeq/

Preseq - http://smithlabresearch.org/software/preseq/

Quicksect - https://github.com/brentp/quicksect

Databases

Ensembl hg19 - http://www.ensembl.org/Homo_sapiens/Info/Index

Human ribosomal DNA complete repeating unit -

http://www.ncbi.nlm.nih.gov/nuccore/U13369

GTF formatted file for repeat annotations-

http://labshare.cshl.edu/shares/mhammelllab/www-

data/TEToolkit/TE_GTF/hg19_rmsk_TE.gtf.gz

Repeat elements (*RepBase20.07*) – http://www.girinst.org/repbase/

V(D)J genes of *B and T cell receptor* - http://www.imgt.org/vquest/refseqh.html#V-D-J-

C-sets

Database of viral genomes: http://ftp.ncbi.nlm.nih.gov/genomes/Viruses

Database of bacterial genomes: http://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/

Database of eukaryotic pathogens - http://eupathdb.org/eupathdb/

References:

1. Jin, Y., Tam, O. H., Paniagua, E. & Hammell, M. TEtranscripts: a package for including transposable elements in differential expression analysis of RNA-seq datasets. *Bioinformatics* btv422 (2015).
2. Melé, M. *et al.* The human transcriptome across tissues and individuals. *Science (80-.).* **348,** 660–665 (2015).
3. Anders, S., Pyl, P. T. & Huber, W. HTSeq--A Python framework to work with high-throughput sequencing data. *Bioinformatics* btu638 (2014).
4. Tarailo-Graovac, M. & Chen, N. Using RepeatMasker to identify repetitive elements in genomic sequences. *Curr. Protoc. Bioinforma.* 4–10 (2009).
5. Bunge, J. & Fitzpatrick, M. Estimating the number of species: a review. *J. Am. Stat. Assoc.* **88,** 364–373 (1993).
6. Deng, C., Daley, T. & Smith, A. D. Applications of species accumulation curves in large-scale biological data analysis. *J. Quant. Biol.*
7. Daley, T. & Smith, A. D. Predicting the molecular complexity of sequencing libraries. *Nat. Methods* **10,** 325–327 (2013).
8. Daley, T. P. Non-parametric Models for Large Capture-recapture Experiments with Applications to DNA Sequencing. (University of Southern California, 2014).
9. Good, I. J. The population frequencies of species and the estimation of population parameters. *Biometrika* **40,** 237–264 (1953).

10. Favaro, S., Lijoi, A. & Prünster, I. A new estimator of the discovery probability. *Biometrics* **68,** 1188–1196 (2012).

11. Colwell, R. K. & Coddington, J. A. Estimating terrestrial biodiversity through extrapolation. *Philos. Trans. R. Soc. B Biol. Sci.* **345,** 101–118 (1994).

12. Willis, A., Bunge, J. & Whitman, T. Inference for changes in biodiversity. *arXiv Prepr. arXiv1506.05710* (2015).

13. Mao, C. X. Predicting the conditional probability of discovering a new class. *J. Am. Stat. Assoc.* **99,** (2004).