Chapter 17

Multimodal T Cell Analysis with CoNGA

Stefan A. Schattgen, William D. Hazelton, Paul G. Thomas, and Philip Bradley

Abstract

Advances in single-cell technologies have made it possible to simultaneously quantify gene expression and immune receptor sequence across thousands of individual T or B cells in a single experiment. Data from such experiments are advancing our understanding of the relationship between adaptive immune receptor sequence and transcriptional profile. We recently reported a software tool, CoNGA, specifically developed to detect correlation between receptor sequence and transcriptional profile. Here we describe in detail how CoNGA can be applied to analyze a large and diverse T cell dataset featuring multiple donors and batch annotations. Our workflow illustrates new analysis modes for the detection of TCR sequence convergence into similarity clusters and of matches to literature-derived TCR databases, as well as processing of gamma-delta T cells.

Key words TCR, CoNGA, Gene expression, TCR convergence

1 Introduction

Advances in single-cell technologies have made it possible to simultaneously quantify gene expression and other modalities (e.g., surface protein abundance, chromatin accessibility, and immune receptor sequence) across thousands of individual cells in a single experiment. These technologies are being applied by immunologists to study T and B cells in a wide range of biological contexts including cancer [1–8], infectious disease [9], and homeostasis [10]. We recently reported a software tool, CoNGA, for the analysis of multimodal single-cell data on T cells [11]. The primary goal of CoNGA is to identify correlations between T cell receptor (TCR) sequence and transcriptional profile. Here we provide a detailed workflow for applying CoNGA that illustrates the latest developments to the software in the context of the reanalysis of a large and multifaceted dataset on colitis induced by immune checkpoint blockade [12].

Huang Huang and Mark M. Davis (eds.), *T-Cell Repertoire Characterization*, Methods in Molecular Biology, vol. 2574, https://doi.org/10.1007/978-1-0716-2712-9_17,

CoNGA identifies correlation between TCR sequence and transcriptional profile by constructing two similarity graphs, one defined by TCR sequence and one by gene expression profile, and looking for a statistically significant overlap between them. These graphs are defined at the level of clonotypes (groups of clonally related cells defined by shared receptor sequence) as opposed to individual cells in order to concentrate on phenotypic relationships correlated with receptor sequence similarity rather than clonal lineage. As originally described, CoNGA has two major modes of analysis: graph-vs-graph analysis and graph-vs-feature analysis. In graph-vs-graph analysis, edges shared between the GEX and TCR similarity graphs are used to identify clusters of clonotypes defined by common features in both modalities. This mode of analysis can identify cell subsets defined by shared TCR sequence features and transcriptional profiles. In graph-vs-feature analysis, numerical features defined by one modality (gene expression or TCR sequence) are projected onto the similarity graph defined by the other modality and graph neighborhoods with biased feature distributions are identified. This mode of analysis can identify an individual gene whose expression is localized within specific regions of the TCR sequence landscape, for example, or TCR sequence features, such as charge or hydrophobicity, that are biased within transcriptionally defined T cell subsets. In the pipeline presented here, we illustrate both of these modes of analyses as well as two additional protocols: one aimed at detecting statistically significant sequence matches to TCRs of known epitope specificity from the literature and one for identifying convergent clusters of TCR sequences that are more similar than would be expected by chance, providing possible evidence of shared antigen specificity. We also demonstrate how to incorporate batch assignments, such as donor origin and disease context, into CoNGA analyses.

A brief outline of the method is as follows. Single-cell transcript count matrices and TCR sequence files are downloaded from the GEO database for this dataset (GSE144469). TCR sequence data for alpha-beta and gamma-delta T cells are processed to define a filtered set of reliable TCR clonotypes, each mapped to a set of one or more cell barcodes. The transcript data and TCR sequence data are then aligned and integrated for analysis with CoNGA. Each cell is assigned donor and disease batch identifiers for annotation of the CoNGA outputs. After standard single-cell preprocessing, the dataset is reduced to a single cell per TCR clonotype, and the gene expression and TCR sequence similarity graphs are constructed. Statistically significant matches to literature TCRs are identified. Graph-vs-graph and graph-vs-feature analyses are performed. Finally, convergent TCR sequence clusters are identified. The above workflow is illustrated in detail for the alpha-beta T cells by

providing the Python commands that would be used in a Jupyter Notebook format [13]; for the gamma-delta T cells, we provide a single shell command for performing the same steps from the terminal using a Python script.

2 Materials

CoNGA Software Comprehensive instructions for installing the CoNGA package and required dependencies are available in the README file attached to the CoNGA GitHub repository (https://github.com/phbradley/conga#readme). We highly recommend using a Python virtual environment for managing the CoNGA installation.

Dataset The dataset on immune checkpoint blockade-induced colitis [12] that we will use here is publicly available from the GEO data repository (GSE144469, https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE144469&format=file). The uploaded data were generated by the authors using the 10× Genomics Single Cell Immune Profiling assay and processed using Cell Ranger software.

3 Methods

Here we apply CoNGA to analyze a large dataset of T cells from a study investigating the origins of colitis as a side effect of immune checkpoint blockade therapy for cancer [12]. This dataset includes cells from 22 donors: 8 receiving checkpoint blockade therapy who developed colitis, 6 receiving checkpoint blockade who did not develop colitis, and 8 healthy controls. We assume that the reader has installed the required Python packages as described above and downloaded the dataset from the GEO database. The code blocks presented below are available in a Jupyter Notebook in the CoNGA GitHub repository [https://github.com/phbradley/conga/blob/7e152e4b0e3311c7e0e230d7bf864c8f35a9af54/fancy_conga_pipeline_with_batches_and_gammadelta_tcrs.ipynb].

3.1 Import the Necessary Python Packages and Navigate to the Folder Containing the Dataset

It will be necessary to change the CONGA_PATH and DATA_DIR variables below to point to the location of the CoNGA GitHub repository and the downloaded dataset, respectively, on the reader's file system.

```
%matplotlib inline
import matplotlib.pyplot as plt
import os
import sys
import scanpy as sc
import numpy as np
import pandas as pd
# path to CoNGA repository (downloaded from github)
CONGA_PATH = '/home/pbradley/gitrepos/conga/'
sys.path.append(CONGA_PATH)
import conga
# path to dataset (downloaded from GEO database)
DATA_DIR = '/home/pbradley/csdat/mimb/data/'
os.chdir(DATA_DIR)
```

3.2 Preprocess the Dataset for CoNGA Analysis

To perform analyses with CoNGA, the barcode assigned to an individual cell in the transcript count matrix and in the TCR sequence file must agree (for cells with both data assignments; *see* **Note 1** for exporting GEX data from Seurat). This will generally be the case for individual datasets, for example, from the 10× genomics chromium assay; however, it may not be true for public datasets downloaded from the Internet or when concatenating multiple datasets. For example, in the alpha-beta T cell data from the GSE144469 dataset analyzed here, the transcript data are stored in individual folders, one per donor, and the cell barcodes all have the same "-1" suffix, whereas the TCR sequence information is provided in a single file in which the cell barcodes have been modified to replace the "-1" suffix with a unique string identifying each donor. For the gamma-delta T cell data, on the other hand, the TCR sequence data are provided in separate files for each donor, and the cell barcodes for the transcript and TCR information both use the "-1" suffix. As the steps required for this preprocessing are somewhat dataset-specific, we provide the details for the GSE144469 dataset in **Appendix 1**. This appendix also illustrates how batch assignments can be added when concatenating multiple datasets. The required preprocessing for the simple case of a single 10× genomics dataset is illustrated in the "simple_conga_pipeline" Jupyter Notebook provided in the CoNGA GitHub repository (https://github.com/phbradley/conga/blob/master/simple_conga_pipeline.ipynb).

In addition to aligning the transcript and TCR cell barcodes, preprocessing involves filtering of the TCR sequence information to eliminate spurious alpha-beta pairings due to promiscuous high-frequency chains (*see* **Note 2**) and (optional) calculation of the kernel principal components (kPCs) from the TCRdist distance matrix of the filtered clonotypes. Since this merged dataset is rather large, we elected not to perform the kPCA calculation, relying instead on the raw TCRdist values to define the TCR neighbor graphs, clusters, and landscape projections (*see* **Note 3**).

3.3 Read the Processed Dataset into Memory

Here we read the merged gene expression (GEX) and TCR sequence information generated in **Appendix 1** into an AnnData Python object (assigned the name "adata" below). The AnnData class (https://anndata.readthedocs.io/en/latest/anndata.AnnData.html) is the fundamental dataset class used by the scanpy [14] single-cell analysis platform on which CoNGA is built. It has standardized locations for storing count matrices (adata.X), one-dimensional cell (adata.obs) and gene (adata.var) features, higher-dimensional cell features (adata.obsm), and unstructured data (adata.uns). In the code block below we use adata.uns to communicate to CoNGA the type of TCR chain (adata.uns['organism']) and the column names of integer-valued batch identifiers (adata.uns['batch_keys']) that should be tracked in the analyses. The per-cell batch information is stored in the corresponding columns of adata.obs.

```
# this string will be prepended to all CoNGA output files
out_prefix = 'conga_abtcr'

# Read the merged AnnData object containing transcript counts and TCR information
# this was generated using the (dataset-specific) pipeline provided in Appendix 1
adata = sc.read_h5ad('merged_gex_abtcr.h5ad')

# Store the organism for this analysis in the uns array
# Set organism to 'human' for human alpha-beta TCR analysis
# other options are 'human_gd' 'human_ig' 'mouse' 'mouse_gd'
adata.uns['organism'] = 'human'

# tell CoNGA to annotate results with these batch definitions (present in adata.obs)
# disease_int: 0= +CPI colitis, 1= +CPI no colitis, 2= healthy control
adata.uns['batch_keys'] = ['disease_int','donor_int']

print('adata.shape:', adata.shape)
print('adata.obs_keys():', adata.obs_keys())
```

```
adata.shape: (60655, 33538)
adata.obs_keys(): ['va', 'ja', 'cdr3a', 'cdr3a_nucseq', 'vb', 'jb', 'cdr3b', 'cdr3b_nucseq', 'diseas
e', 'disease_int', 'donor', 'donor_int', 'batch']
```

From the output above we can see that adata contains information on the expression of 33,538 genes in 60,655 T cells with paired alpha-beta TCR sequences. The TCR sequence information is stored in the first eight columns ('keys') of adata.obs ('va' to 'cdr3b_nucseq'). The remaining adata.obs columns contain batch information in human-readable and integer-valued formats.

3.4 Filter, Scale, and Subset the Dataset; Reduce to Clonotypes; Cluster and Perform Dimensionality Reduction

After loading the merged dataset containing GEX and TCR information into memory, we first perform standard single-cell quality control (QC) and filtering to remove genes that occur in only a few cells and cells with too few or too many expressed genes or too great a fraction of transcripts coming from mitochondrial genes. The count matrix (adata.X) is restricted to a set of highly variable

genes, log-transformed, and scaled (note that the full count matrix is retained in adata.raw.X). We then partition the dataset based on CD4 versus CD8 expression, taking only the CD8 T cells. As reported in the original CoNGA paper [11], consistent differences between the TCR sequences of CD4 and CD8 cells can lead to GEX/TCR correlations that are detected by graph-vs-graph and graph-vs-feature analyses. While these are genuine signals, we have found that their presence can mask other signals of interest, particularly in larger datasets. Subsetting to CD4 or CD8 cells thus helps to focus on correlations of greater potential biological interest (*see* **Note 4**). After subsetting to the CD8 T cells, we reduce to a single representative cell per clonotype (the cell with the smallest average GEX distance to the other cells in the clonotype; *see* **Note 5**) and perform clustering and dimensionality reduction.

```
# Perform QC filtering and scale GEX
adata = conga.preprocess.filter_and_scale(
    adata,
    min_genes_per_cell = 500,
    max_genes_per_cell = 3000,
    max_percent_mito = 0.1
)

# reduce to CD8 cells (see Note 4)
adata_cd4, adata_cd8 = conga.devel.split_into_cd4_and_cd8_subsets(
    adata, verbose= True)
adata = adata_cd8

# Reduce dataset down to a single representative cell per clonotype.
# OPTION: The average_clone_gex option allows you represent GEX of the
#  clonotype as the average across all cells in the clonotype.
adata = conga.preprocess.reduce_to_single_cell_per_clone( adata )

# Perform dimensionality reduction and clustering of GEX and TCR data.
# UMAP reduction and louvain clustering are the defaults.
adata = conga.preprocess.cluster_and_tsne_and_umap(adata)

print('adata.shape:', adata.shape)
print('adata.obs_keys():', adata.obs_keys())
print('adata.obsm keys():', adata.obsm keys())
```

```
adata.shape: (8585, 682)
adata.obs_keys(): ['va', 'ja', 'cdr3a', 'cdr3a_nucseq', 'vb', 'jb', 'cdr3b', 'cdr3b_nucseq', 'diseas
e', 'disease_int', 'donor', 'donor_int', 'batch', 'n_genes', 'percent_mito', 'n_counts', 'leiden_gex
_for_cd4_vs_cd8', 'cd4_or_cd8', 'clone_sizes', 'gex_variation', 'louvain_gex', 'clusters_gex', 'clus
ters_tcr']
adata.obsm_keys(): ['X_pca_gex', 'disease_int', 'donor_int', 'X_umap_gex', 'X_gex_1d', 'X_gex_2d', '
X_tcr_2d', 'X_tcr_1d']
```

From the shape of the adata object we can see that the dataset has been reduced to 8585 cells (one per CD8 clonotype) and 682 highly variable genes. The results of clustering and dimensionality reduction are stored in adata.obs and adata.obsm: the GEX and TCR cluster assignments in the adata.obs columns 'clusters_gex' and 'clusters_tcr', the GEX principal component projections in adata.obsm['X_pca_gex'], and the GEX and TCR 2D UMAP [15] projections in adata.obsm['X_gex_2d'] and adata.obsm

['X_tcr_2d']. Note that if we had performed TCRdist kernel PCA during preprocessing, the TCR principal component information would be stored in adata.obsm['X_pca_tcr'].

3.5 Compute the Neighbor Graphs

In the CoNGA GEX and TCR neighbor graphs, each clonotype is connected to its K-nearest neighbors in GEX and TCR space, respectively. Multiple neighbor graphs can be constructed, each defined by the neighbor fraction K/N where N is the total number of clonotypes. By default, neighbor fractions of 0.01 and 0.1 are used. During neighbor graph construction, a nearest-neighbor distance ('NNdist') is computed for each clonotype in both GEX and TCR space; this weighted average of the distance to its K-nearest neighbors provides a measure of local (inverse) density nearby that clonotype for use in subsequent analyses.

```
# specify the neighbor fractions
nbr_fracs = [0.01, 0.1]

# construct the K nearest neighbor graphs for each neighbor fraction
all_nbrs, nndists_gex, nndists_tcr = conga.preprocess.calc_nbrs(
    adata, nbr_fracs, also_calc_nndists=True,
    nbr_frac_for_nndists= nbr_fracs[0],
)

# store the NNdists in adata.obs
adata.obs['nndists_gex'] = nndists_gex
adata.obs['nndists_tcr'] = nndists_tcr

# stores the annotated TCR cluster names in adata.uns
conga.preprocess.setup_tcr_cluster_names(adata)
```

3.6 Match to a Database of Literature-Derived TCR Sequences

In this step we look for sequence matches to TCRs of known epitope specificity described in the literature [16–21]. For paired-chain matches, the raw TCRdist score for each match is converted to a probability that takes into account how far from germline the matched sequences are. This probability is adjusted to account for the number of TCRdist comparisons being performed, and an FDR value is computed. For single-chain matches, all exact CDR3 sequence matches are reported. The top 15 paired-chain matches are reported in Table 1.

```
# match TCRs in the dataset to a database derived from literature sources
match_results = conga.tcr_clumping.match_adata_tcrs_to_db_tcrs(
    adata, outfile_prefix= out_prefix,
    num_random_samples_for_bg_freqs=500000, #default is 50000
)

# There is also the ability to simply look for single chains matching in amino acid sequence
alpha_hits, beta_hits = conga.tcr_clumping.strict_single_chain_match_adata_tcrs_to_db_tcrs(
    adata, outfile_prefix = out_prefix,
)

# show the top paired results, focusing on a few columns of interest
cols = 'tcrdist pvalue_adj va cdr3a vb cdr3b db_epitope db_epitope_gene db_mhc_trim'.split()
match_results.drop_duplicates('clone_index')[cols].head(15)
```

Table 1
Top 15 abTCR database matches

TCRdist	*P* value	va	cdr3a	vb	cdr3b	Epitope	Source	Gene	MHC
0	3.54E-05	TRAV5*01	CAEYSSASKIIF	TRBV14*01	CASSQSPGGTQYF	GLCTLVAML	EBV	BMLF1	A*02
0	3.54E-05	TRAV5*01	CAEYSSASKIIF	TRBV14*01	CASSQSPGGTQYF	GLCTLVAML	EBV	BMLF1	A*02
0	3.54E-05	TRAV5*01	CAESIGKLIF	TRBV29–1*01	CSVGTGGTNEKLFF	GLCTLVAML	EBV	BMLF1	A*02
0	3.54E-05	TRAV17*01	CATVLRMDSSYKLIF	TRBV7–9*01	CASSLIGVSSYNE QFF	TP RVTGGGAM	CMV	pp65	B*07
6	3.54E-05	TRAV5*01	CAESIGKIIF	TRBV29–1*01	CSVGTGGTNEKLFF	GLCTLVAML	EBV	BMLF1	A*02
9	3.54E-05	TRAV5*01	CASDDNARLMF	TRBV20–1*01	CSARDQTGNGYTF	GLCTLVAML	EBV	BMLF1	A*02
24	3.54E-05	TRAV5*01	CADDFNARLMF	TRBV20–1*01	CSARDRGLGNTIYF	GLCTLVAML	EBV	BMLF1	A*02
0	0.000849709	TRAV8–6*01	CAVGGSQGNLIF	TRBV19*01	CASSIRSSYEQYF	GILGFVFTL	FLU	M1	A*02
0	0.000991327	TRAV5*01	CAESTGKLIF	TRBV29–1*01	CSVGTGGTNEKLFF	GLCTLVAML	EBV	BMLF1	A*02
0	0.000991327	TRAV5*01	CAESTGKLIF	TRBV29–1*01	CSVGTGGTNEKLFF	GLCTLVAML	EBV	BMLF1	A*02
0	0.000991327	TRAV5*01	CAESTGKLIF	TRBV29–1*01	CSVGTGGTNEKLFF	GLCTLVAML	EBV	BMLF1	A*02
24	0.001982654	TRAV17*01	CATEGNSGYSTLTF	TRBV6–5*01	CASSTQGGGRGYTF	FLYALALLL	EBV	LMP2A	A*02
0	0.002265891	TRAV26–2*01	CILPLAGGTSYGKL TF	TRBV7–8*01	CASSLGQAYEQYF	FLRGRAYGL	EBV	EBNA3A	B*08
0	0.002265891	TRAV26–2*01	CILPLAGGTSYGKL TF	TRBV7–8*01	CASSLGQAYEQYF	FLRGRAYGL	EBV	EBNA3A	B*08
24	0.00382369	TRAV5*01	CAESTPRGKLIF	TRBV29–1*01	CSVGTGGTNEKLFF	GLCTLVAML	EBV	BMLF1	A*02

3.7 Run Graph-vs-Graph Analysis

CoNGA graph-vs-graph analysis identifies clonotypes whose neighbors in GEX space overlap significantly with their neighbors in TCR space. These clonotypes are grouped into CoNGA clusters based on their joint GEX and TCR cluster assignments, and a logo-style visualization is produced for each cluster (Fig. 1). GEX landscapes colored by GEX-neighbor-averaged marker gene expression help to delineate T cell subsets. Here we are using the 0.1 neighbor fraction graph for averaging; this produces fairly smooth landscapes, whereas a smaller neighbor fraction would preserve more of the fine-scale variation that can be seen in the raw expression plots. As illustrated in the code block below, CoNGA analyses are typically divided into two function calls, one that performs the calculations and stores the results in the AnnData object and a second that produces the visualizations.

```
# Run the graph-vs-graph analysis and store results in adata.uns['conga_results']
conga.correlations.run_graph_vs_graph(adata, all_nbrs, outfile_prefix = out_prefix)

# Specify the nbr fraction used for averaging GEX and TCR features for plotting
nbrs_gex, nbrs_tcr = all_nbrs[0.1]

# Specify the minimal conga cluster size for plotting.
# We typically use 0.001 * number of clonotypes in the dataset to limit the FDR.
min_cluster_size = 8

# Generate graph-vs-graph results plot
# If batch keys have been specified and database matching results are stored in the object,
#  they will appear in the plot by default.
conga.plotting.make_graph_vs_graph_logos(
    adata, out_prefix, min_cluster_size,
    nbrs_gex,
    nbrs_tcr,
)
```

From the batch-assignment visualizations in Fig. 1 (the colorful stacked bars labeled 'disease_int' and 'donor_int'), we can see that there are several CoNGA clusters found exclusively or almost exclusively in the colitis-positive donors (the bottom cluster and the top 11 clusters; note the tall blue segments in the 'disease_int' bars). On the other hand, the mucosal-associated invariant T (MAIT) cell clusters— (10, 6) and (10, 4)—appear somewhat depleted within the colitis-positive subset.

3.8 Run Graph-vs-Feature Analyses

In graph-vs-feature analysis, features defined by one modality (gene expression or TCR sequence) are projected onto the neighbor graph defined by the other modality in order to identify graph neighborhoods with biased feature distributions. This analysis can identify genes whose expression is localized to specific TCR sequence clusters and CDR3 biochemical features such as charge or hydrophobicity that are significantly biased within T cell subsets defined by gene expression. Since the original development and publication of the CoNGA algorithm, we have implemented the Yosef Lab's HotSpot algorithm [23] as an additional approach for identifying these features. HotSpot can in some cases be more

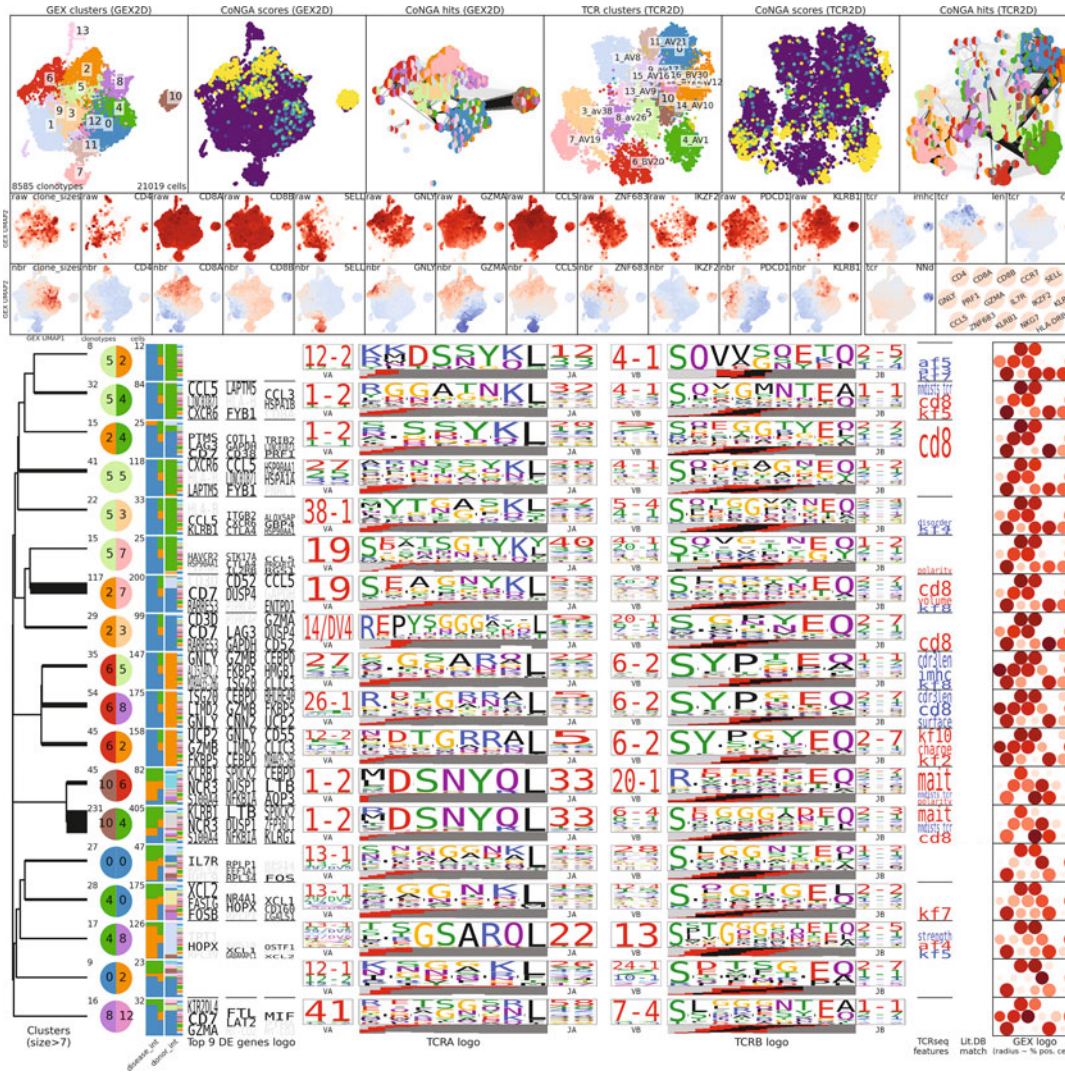

Fig. 1 Graph-vs-graph logo plots. At the top are 2D UMAP projections of clonotypes in the dataset based on GEX similarity (left three panels) and TCR similarity (right three panels), colored from left to right by GEX cluster assignment; CoNGA score; joint GEX-TCR cluster assignment for clonotypes with significant CoNGA scores, using a bicolored disk whose left half indicates GEX cluster and whose right half indicates TCR cluster; TCR cluster; CoNGA score; and GEX-TCR cluster assignments for CoNGA hits. Below are two rows of smaller GEX landscapes colored by selected marker genes (left) and TCR sequence features (right); for genes, raw counts are shown as well as neighbor-averaged Z scores. Finally, the GEX and TCR sequence features of CoNGA clusters containing eight or more clonotypes are summarized by a series of logo-style visualizations, from left to right, cluster dendrogram based on KNN graph connections; bicolored disk showing the GEX and TCR cluster assignments; stacked bar plots showing batch distribution of clustered cells; differentially expressed genes (DEGs) and TCR sequence logos showing V and J gene usage and CDR3 sequences [22]; biased TCR sequence scores, with red indicating elevated scores and blue indicating decreased scores; gene expression dot plots showing mean expression levels and fraction of expressing cells for a panel of marker genes (gene names shown in the panel above). DEG and TCR sequence logos are scaled by the adjusted P value of the associations, with full logo height requiring a top adjusted P-value below 10^{-6}. DEGs with fold changes less than 2 are shown in gray. The stacked bars showing the batch composition of each CoNGA cluster are split vertically, with the left, thicker portion showing the batch composition of the cells in that cluster and the right portion showing the batch composition of all the cells in the dataset

sensitive than our original graph-vs-feature analysis, since it assesses feature/graph correlation across the entire graph rather than looking at the feature score distribution in each clonotype's neighborhood individually. On the other hand, features that are highly elevated in a very small subpopulation may receive a more significant score in the original CoNGA graph-vs-feature analysis. We sometimes find that neighbor graphs with small neighborhoods (e.g., neighbor fractions <0.02) give less interpretable/robust results; here we are using only the 0.1 neighbor fraction (i.e., the K-nearest neighbor graph with K = 0.1 * num_clonotypes) for the HotSpot analysis. Figure 2 shows a heatmap of the top 50 nonredundant features from HotSpot graph-vs-feature analysis.

```
# run the graph-vs-features analysis and store results
#  in adata.uns['conga_results']
conga.correlations.run_graph_vs_features(
    adata, all_nbrs, outfile_prefix = out_prefix)

# generate plots visualizing the results
conga.plotting.make_graph_vs_features_plots(
    adata, all_nbrs, out_prefix,
    clustermap_max_type_features=25,
)

# run an alternative graph-vs-features analysis
#  using the Yosef lab's HotSpot algorithm
conga.correlations.find_hotspots_wrapper(
    adata, all_nbrs, nbr_fracs = [0.1],
    outfile_prefix = out_prefix,
)

# generate plots visualizing the results
conga.plotting.make_hotspot_plots(
    adata, all_nbrs,
    outfile_prefix = out_prefix,
    clustermap_max_type_features=25,
)
```

3.9 Look for Evidence of TCR Sequence Convergence

TCR sequence clustering above and beyond what would be expected under a null model of VDJ rearrangement can signal recurrent selection of T cell clonotypes responding to the same epitope. CoNGA's "TCR clumping" algorithm detects sequence convergence by matching the TCR sequences to each other and to a large set of shuffled TCR sequences and then looking for TCR neighborhoods in the dataset that are overpopulated relative to background expectation. The clonotypes with enriched neighborhoods are grouped into clusters and these clusters are visualized by logo plots similar to those for the graph-vs-graph results (Fig. 3). Here we can see MAIT cells (cluster 1) and a large cluster (cluster 2) of clonotypes characterized by usage of the TRBV6–2 gene segment. This cluster, which comes from a single donor (orange in the donor_int batch bar), can also be seen in the graph-vs-graph logo plot (Fig. 1).

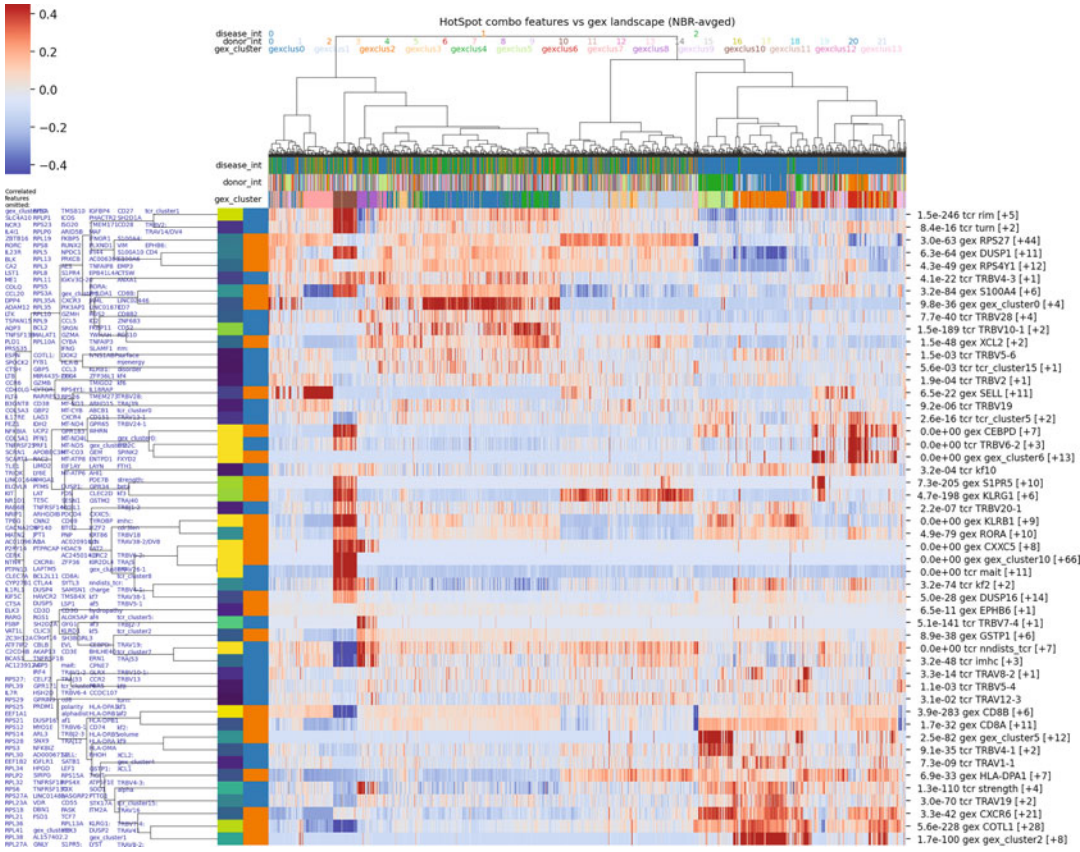

Fig. 2 HotSpot graph-vs-feature analysis. HotSpot analysis identified 358 GEX features with biased distributions over the TCR neighbor graph and 79 TCR features with biased distributions across the GEX neighbor graph. Hierarchical clustering was used to reduce this to 25 nonredundant features for each modality, which are visualized here across the GEX landscape using a heatmap colored by neighbor-averaged feature Z-scores. Rows correspond to features; columns represent clonotypes. The rows (features) are ordered sequentially by hierarchical clustering of the feature values as indicated in the dendrogram to the left of the heatmap; the columns (clonotypes) are ordered by hierarchical clustering of the GEX principal component landscape. The three rows above the heatmap show the disease and donor batch assignments and the GEX cluster number, as indicated. The two columns to the left of the heatmap show the feature type (orange = GEX, blue = TCR) and adjusted *P*-value for each feature. The remaining redundant/correlated features that associate with the 50 representative features are shown along the left of the image

```
# Run TCR clumping analysis
conga.tcr_clumping.assess_tcr_clumping(
    adata, outfile_prefix = out_prefix,
)

# make plots to visualize the results
conga.plotting.make_tcr_clumping_plots(
    adata, nbrs_gex, nbrs_tcr, out_prefix,
    min_cluster_size_for_logos=8,
)
```

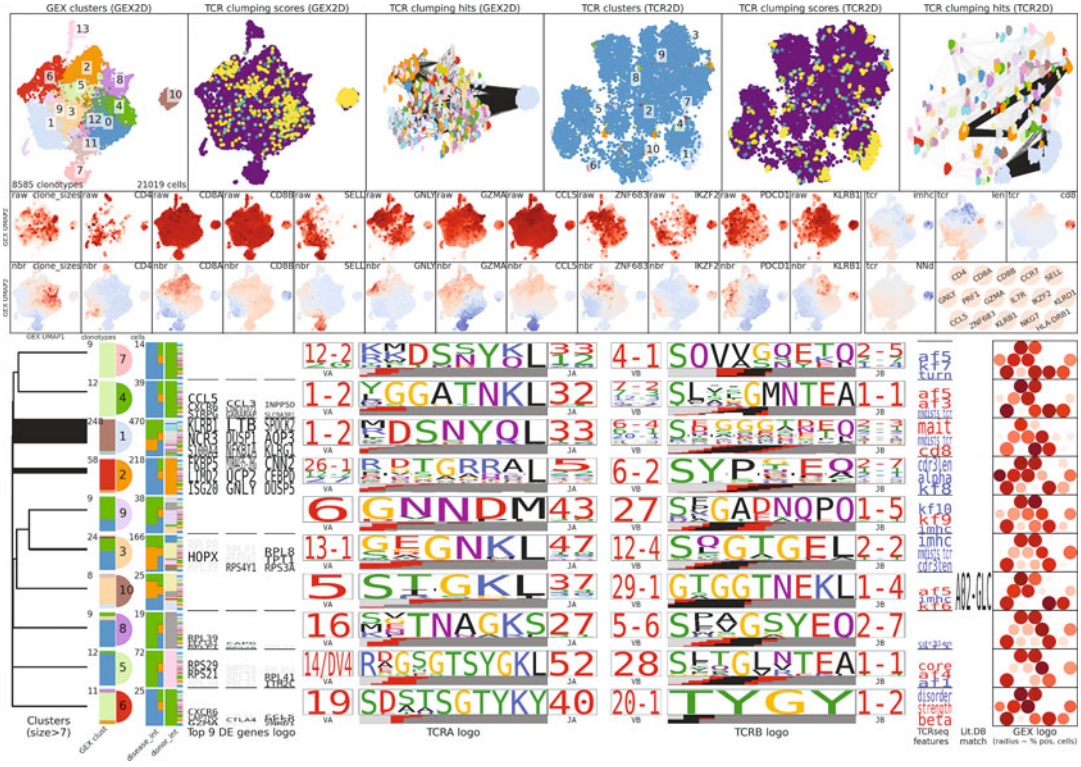

Fig. 3 TCR sequence convergence logo plots. TCR clonotype clusters detected by CoNGA's TCR clumping analysis are visualized using logo plots in the same layout as in Fig. 1, with the exception that each cluster is represented by a stacked bar immediately to the right of the dendrogram indicating the GEX cluster distribution of the cluster members. Matches to literature-derived TCR sequences of known specificity identified in **Step 3.6** above are displayed in the second to last column on the right, where we can see that sequence cluster 10 has significant matches to the HLA A*02-presented GLC peptide (derived from the EBV BMLF1 antigen)

3.10 Review Results Cached in the AnnData Object

The results of these analyses are stored as Pandas [24] DataFrames in the adata.uns['conga_results'] dictionary, as are links to image file locations and help messages describing the tables and figures generated by CoNGA. A set of summary statistics are stored in adata.uns['conga_stats'].

```
# print all the results that have been stored in adata.uns['conga_results']
print('conga results keys:', ' '.join(adata.uns['conga_results'].keys()))

# show the stats that are stored in adata.uns['conga_stats']
adata.uns['conga_stats']
```

```
conga results keys: tcr_db_match tcr_db_match_help graph_vs_graph graph_vs_graph_help graph_vs_graph_log
os graph_vs_graph_logos_help tcr_graph_vs_gex_features tcr_graph_vs_gex_features_help tcr_genes_vs_gex_f
eatures tcr_genes_vs_gex_features_help gex_graph_vs_tcr_features gex_graph_vs_tcr_features_help graph_vs
_features_gex_clustermap graph_vs_features_gex_clustermap_help graph_vs_features_tcr_clustermap graph_vs
_features_tcr_clustermap_help tcr_graph_vs_gex_features_plot tcr_graph_vs_gex_features_plot_help tcr_gra
ph_vs_gex_features_panels tcr_graph_vs_gex_features_panels_help tcr_genes_vs_gex_features_panels tcr_gen
es_vs_gex_features_panels_help gex_graph_vs_tcr_features_plot gex_graph_vs_tcr_features_plot_help gex_gr
aph_vs_tcr_features_panels gex_graph_vs_tcr_features_panels_help hotspot_features hotspot_features_help
hotspot_gex_umap hotspot_gex_umap_help hotspot_gex_clustermap hotspot_gex_clustermap_help hotspot_tcr_um
ap hotspot_tcr_umap_help hotspot_tcr_clustermap hotspot_tcr_clustermap_help tcr_clumping tcr_clumping_he
lp tcr_clumping_logos tcr_clumping_logos_help

OrderedDict([('min_genes_per_cell', 500),
             ('max_genes_per_cell', 3000),
             ('max_percent_mito', 0.1),
             ('num_filt_max_genes_per_cell', 4650),
             ('num_filt_max_percent_mito', 1914),
             ('num_antibody_features', 0),
             ('num_TR_genes', 171),
             ('num_TR_genes_in_hvg_set', 89),
             ('num_highly_variable_genes', 682),
             ('num_cells_after_filtering', 54064),
             ('num_clonotypes', 8585),
             ('max_clonotype_size', 336),
```

3.11 Save the Final AnnData Object as Well as an HTML-Formatted Summary Webpage

The final AnnData object, including the analysis results cached in adata.uns, can be saved to a file for later reanalysis. A summary web page can also be generated for viewing in a web browser.

```
# save adata object to a file for later analysis
adata_file = out_prefix+'_final.h5ad'
adata.write(adata_file)

# make an .html summary file for viewing in a browser
html_file = out_prefix+'_results_summary.html'
conga.plotting.make_html_summary(adata, html_file)
```

3.12 Perform CoNGA Analysis of the Gamma-Delta T Cells Using the run_conga.py Script

These CoNGA analyses can also be accessed through a single Python script that wraps the code blocks presented above (Figs. 4 and 5). The command below was used to run the full analysis pipeline on the gamma-delta TCRs from this dataset.

```
python /home/pbradley/gitrepos/conga/scripts/run_conga.py \
    --gex_data merged_gex_gdtcr.h5ad \
    --gex_data_type h5ad \
    --outfile_prefix conga_gdtcr \
    --organism human_gd \
    --batch_keys disease_int donor_int \
    --short_clustermaps \
    --no_kpca --all
```

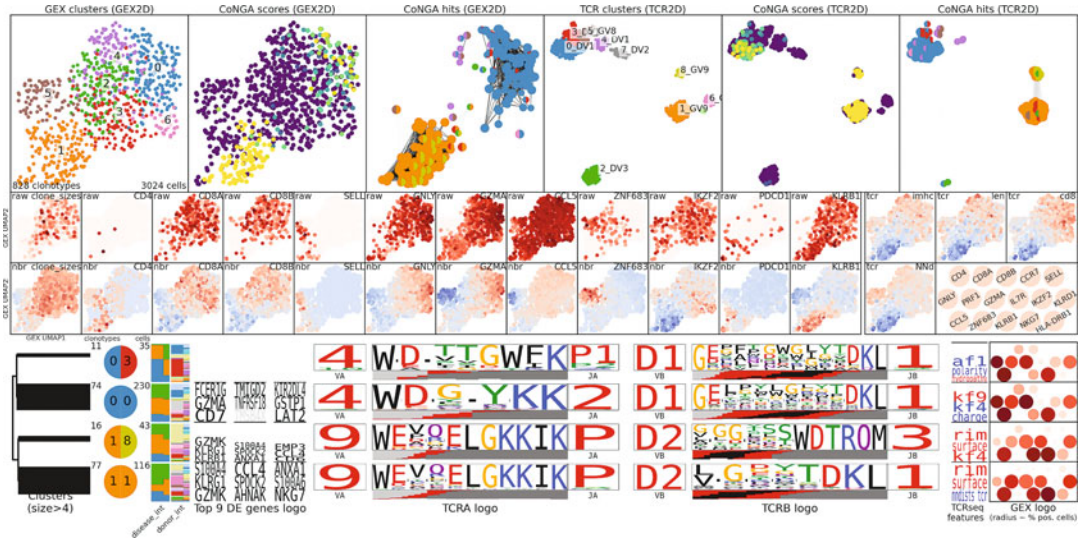

Fig. 4 Graph-vs-graph logo plots for the gdTCRs. Same layout as in Fig. 1; here we can see two large clusters, one featuring a TRGV9/TRDV2 gene pairing and one with a TRGV4/TRDV1 gene pairing

4 Notes

1. Migrating from Seurat to CoNGA

In instances where extensive investigation has been done using other single-cell analysis software, for example, Seurat, or in cases where the original GEX counts matrix is not available and only a Seurat object is accessible, it may be convenient to export the GEX information (and other surface protein information, if available) in a format suitable for import into CoNGA/scanpy. We recommend using the write10× Counts function from the DropletUtils package for exporting the necessary information from Seurat in 10× format.

```
require(Seurat)
require(DropletUtils)

# open Seurat object
hs1 <- readRDS('~/vdj_v1_hs_V1_sc_5gex.rds')
# export GEX counts matrix and feature information in 10X format
write10xCounts(x = hs1@assays$RNA@counts, path = './hs1_mtx/')
# Import the hs1_mtx directory into CoNGA using the '10x_mtx' option.
```

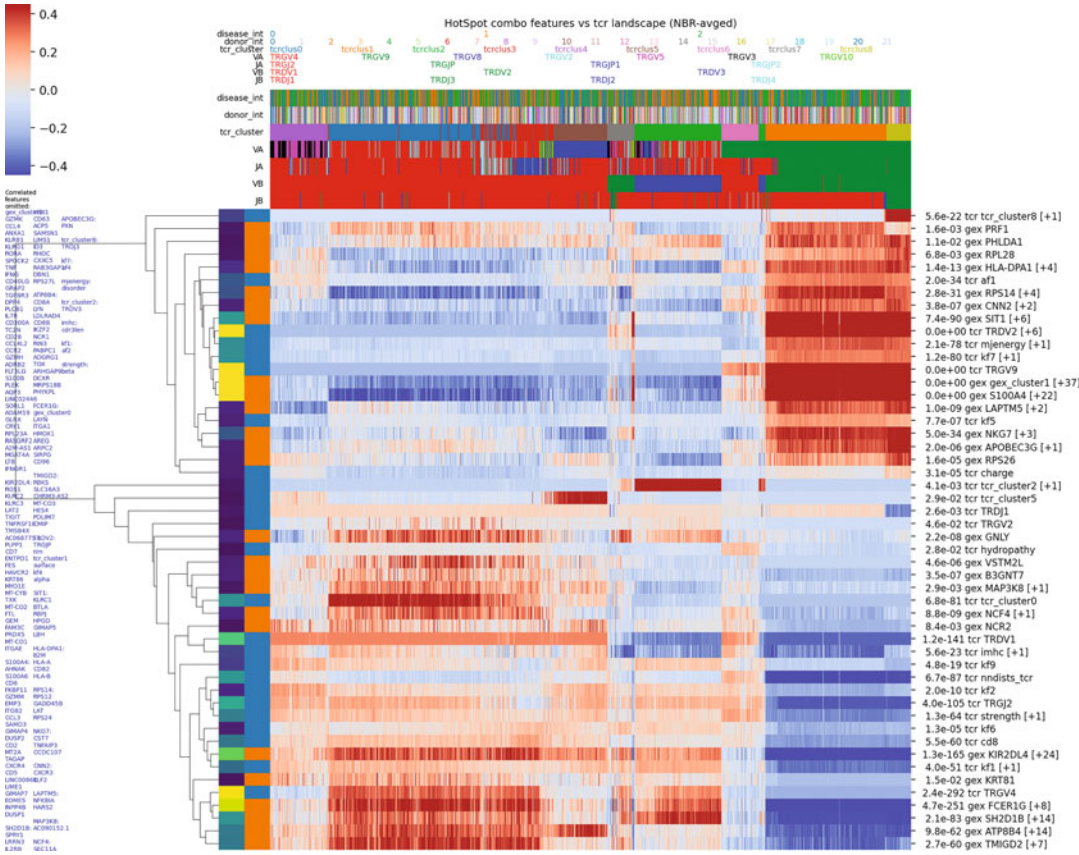

Fig. 5 HotSpot graph-vs-feature analysis for the gdTCRs. Same visualization as Fig. 2 with the difference that the columns (clonotypes) are ordered by TCR sequence similarity, rather than GEX similarity, and the TR gene segments are shown in the four rows immediately above the heatmap. We can see the features associated with the TRGV9/TRDV2 pairing in the top-right corner of the heatmap. Note that the labeling uses the alpha nomenclature, with alpha corresponding to gamma and delta corresponding to beta

If exporting a multimodal dataset with, for example, TotalSeq labeling, we can concatenate the count matrices prior to exporting:

```
# RNA' is the GEX, and 'ADT' is the antibody labeling assay slot.
count_matrix <- rbind(hs1@assays$RNA@counts, hs1@assays$ADT@counts)

# create vector of feature type labels
features <- c(
rep("Gene Expression", nrow(hs1@assays$RNA@counts)),
rep("Antibody Capture", nrow(hs1@assays$ADT@counts))
)

# write out
write10xCounts( count_matrix,
path = './hs1_mtx/',
gene.id = rownames(count_matrix),
gene.symbol = rownames(count_matrix),
barcodes = colnames(count_matrix),
gene.type = features,
version = "3")
# Import the hs1_mtx directory into CoNGA using the '10x_mtx' option.
```

2. Clonotype Filtering

Single-cell TCR sequencing datasets often suffer from promiscuous pairing of high-frequency chains with secondary partners at much lower frequency due to the presence of cell doublets and/or ambient TCR mRNA from ruptured cells. For this reason, CoNGA filters the set of TCR chains recovered from each cell based on the pairings observed in all the other cells in the dataset in order to define a set of reliable clonotypes. The first step is to count the number of times each alpha-beta pairing is observed in the dataset, with each cell contributing a count of 1 for all alpha-beta chain pairs observed in that cell. The set of possible alpha-beta chain pairs is sorted by this count (a measure of the support for that pairing in the dataset) in decreasing order, and alpha-beta pairs are successively added to a list of filtered pairs provided that they do not overlap with a higher-count pairing. Second in-frame alpha chains are allowed provided that their support is at least one-third of the support of the primary alpha-beta pairing. Cells with a single in-frame beta chain and up to two in-frame alpha chains, where all alpha-beta pairings are on the filtered pairing list, are assigned to the clonotype defined by those chains. Cells in which only one of the two in-frame alphas of a valid dual-alpha clonotype was observed are assigned to that clonotype. This stringent clonotype filtering is important because cells derived from the same clonal lineage tend to have similar gene expression profiles; artificially splitting true clonotypes can therefore create apparent GEX/TCR correlation, since the split clonotypes will also share one or the other of their TCR chains and thus have greater than expected TCR sequence similarity in addition to greater than expected GEX similarity.

3. TCRdist Kernel Principal Components Analysis

In the original CoNGA publication, we described a workflow in which the matrix of inter-clonotype TCRdist distances is converted to a reduced dimensionality representation by kernel principal component analysis (kPCA). This had the advantage that the reduced dimensionality representation could be easily incorporated into standard single-cell workflows for clustering and landscape projection by simply replacing the GEX PCA matrix. Focusing on the top principal components in kPCA also allowed for some degree of smoothing of the original high-dimensional TCR sequence landscape. For larger datasets ($>50{,}000$) we have found that kPCA can be fairly CPU- and memory-intensive, leading us to implement an alternative pipeline in which the raw TCRdist distances are used for neighbor-graph construction, clustering, and landscape projection. This is the approach taken in the pipeline presented here. A related issue is that if multiple independently

preprocessed datasets with kPCA projections are concatenated, their kPCA projections will automatically be concatenated by the AnnData machinery, but these projections will not be in the same frame of reference (since each kPCA calculation was carried out independently). Thus, it is important either to rerun the kernel PCA calculation on the full merged dataset, or to delete the 'X_pca_tcr' field from the adata.obsm array and use the raw TCRdist distances in CoNGA.

4. Splitting Mixed T Cell Datasets into CD4+ and CD8+ Subsets

We and others have observed that there are consistent TCR sequence differences between CD4+ and CD8+ T cells: TCRs from CD4+ T cells tend to be more positively charged, for example, and they sample from a different distribution of V and J gene segments than the one used by CD8+ T cells [25–28]. Although these differences are not absolute (it is not possible to predict whether an individual cell is a CD4+ T cell or a CD8+ T cell from its TCR sequence alone), they can combine with the transcriptional differences between CD4+ and CD8+ T cells to generate covariation between TCR and GEX that is detectable by CoNGA analysis. We have found that the presence of this CD4 versus CD8 covariation can make it harder to recognize other signals of TCR/GEX covariation, particularly in larger datasets. For this reason, we recommend splitting datasets by CD4/CD8 status for CoNGA analysis and comparing these results to the analyses of the full, merged datasets. This can be done in a fully automated fashion as in **Step 3.4** above, but human intervention and/or incorporation of surface protein expression may yield better results.

5. Analyzing Clonotypes Versus Single Cells

T cells belonging to the same clonal lineage (here referred to as a clonotype) will all have the same TCR sequence. In addition, we and others have observed that clonally related T cells have higher than average transcriptional similarity. For these reasons, CoNGA neighbor graphs and correlation analyses are conducted at the level of clonotypes rather than individual cells. This necessarily involves the loss of some information, particularly in cases where there is transcriptional diversity within individual clonotypes. By default, the cell with the smallest average GEX distance (measured by Euclidean distance in GEX principal component space) to the other cells in an expanded clonotype is selected as the representative for that clonotype. An alternative is to average the GEX profiles of all the cells in the clonotype and use this as the clonotype's GEX profile; this behavior can be selected by passing average_clone_gex = True to the conga.preprocess.reduce_to_single_cell_per_clone function (*see* **Step 3.4**).

Acknowledgments

This research was supported by NIH R01AI136514, R21AI169085, U01AI150747, the Neoma Boadway Fellowship (SS), and ALSAC at St. Jude.

Appendix 1

In this appendix we provide a detailed description of the preprocessing steps necessary to prepare the GSE144469 dataset for CoNGA analysis. To perform analyses with CoNGA, the barcode assigned to an individual cell in the transcript count matrix and in the TCR sequence file must agree (for cells with both data assignments). This will generally be the case for individual datasets, for example, from the 10× genomics chromium assay; however, it may not be true for public datasets downloaded from the Internet or when concatenating multiple datasets. In the alpha-beta T cell data from the GSE144469 dataset analyzed here, the transcript data are stored in individual folders, one per donor, and the cell barcodes all have the same "-1" suffix, whereas the TCR sequence information is provided in a single file in which the cell barcodes have been modified to replace the "-1" suffix with a unique string identifying each donor. For the gamma-delta T cell data, on the other hand, the TCR sequence data are provided in separate files for each donor, and the cell barcodes for the transcript and TCR information both use the "-1" suffix.

Step 1. Import the necessary Python packages and navigate to the folder containing the dataset.

It will be necessary to change the CONGA_PATH and DATA_DIR variables below to point to the location of the CoNGA GitHub repository and the downloaded dataset, respectively, on the reader's file system.

```
%matplotlib inline
import matplotlib.pyplot as plt
import os
import sys
import glob
import scanpy as sc
import numpy as np
import pandas as pd
# path to CoNGA repository (downloaded from github)
CONGA_PATH = '/home/pbradley/gitrepos/conga/'
sys.path.append(CONGA_PATH)
import conga
from conga.tcrdist.make_10x_clones_file import make_10x_clones_file
DATA_DIR = '/home/pbradley/csdat/mimb/data/'
os.chdir(DATA_DIR)
```

Step 2. Process and merge the abTCR data.

We first read the TCR sequence information from the contig annotations file. Then, for each of the 22 donors we modify the cell barcodes in the TCR data to match the GEX data, filter the TCR clonotypes, merge the GEX and TCR data into a single AnnData object, and assign batch identifiers giving the donor ID and disease status. Finally, a single AnnData object is created by merging the 22 individual datasets and this object is saved to disk for subsequent CoNGA analysis.

```
gex_datasets = sorted(glob.glob('*-CD3/'))
diseases = ['C','NC','CT'] # colitis, no-colitis, healthy control
contigs_file = 'GSE144469_TCR_filtered_contig_annotations_all.csv'
all_contigs = pd.read_csv(contigs_file)
all_data = []
for donor_num, gex_dir in enumerate(gex_datasets):
    # The folder name is also the donor ID
    donor = gex_dir.split('/')[-2].split('-')[0]
    donor_contigs = all_contigs[all_contigs.barcode.str.endswith(donor)].copy()
    # change the barcode suffix to '-1' to match the GEX data
    donor_contigs['barcode'] = donor_contigs.barcode.str.split('-').str.get(0)+'-1'
    donor_contigs_file = f'{donor}_abtcr_filtered_contigs.csv'
    donor_contigs.to_csv(donor_contigs_file)
    # process the contigs to generate conga clonotypes
    donor_clones_file = f'{donor}_abtcr_clones.tsv'
    make_10x_clones_file(
        donor_contigs_file,
        organism = 'human', # using 'human' for TCRab
        clones_file = donor_clones_file,
        stringent = True, # (the default) see Note #1 on clonotype filtering
    )
    # read the GEX data and the clonotypes into CoNGA
    adata = conga.preprocess.read_dataset(
        gex_dir, '10x_mtx', donor_clones_file,
        allow_missing_kpca_file=True)
    disease = donor[:-1]
    adata.obs['disease'] = disease
    adata.obs['disease_int'] = diseases.index(disease) # conga batch ids are integers
    adata.obs['donor'] = donor
    adata.obs['donor_int'] = donor_num # conga batch ids are integers

    all_data.append( adata )

# concatenate the datasets
new_adata = all_data[0].concatenate(all_data[1:])
#save the aggregate AnnData object
new_adata.write('merged_gex_abtcr.h5ad')
```

Step 3. Process and merge the gdTCR data.

Then, for each of the 21 donors with gdTCR data we filter the TCR clonotypes, merge the GEX and TCR data into a single AnnData object, and assign batch identifiers giving the donor ID and disease status. Finally, a single AnnData object is created by merging the 21 individual datasets and this object is saved to disk for subsequent CoNGA analysis.

```
gex_datasets = sorted(glob.glob('*-CD3/'))

diseases = ['C','NC','CT'] # colitis (0), no-colitis (1), healthy control (2)
all_data = []

for donor_num, gex_dir in enumerate(gex_datasets):
    # The folder name is also the donor ID
    donor = gex_dir.split('/')[-2].split('-')[0]
    donor_contigs_file = glob.glob(f'*_{donor}-gdTCR_filtered_contig_annotations.csv')
    if len(donor_contigs_file)==0:
        continue
    else:
        assert len(donor_contigs_file) == 1
        donor_contigs_file = donor_contigs_file[0]
    # process the contigs to generate conga clonotypes
    donor_clones_file = f'{donor}_gdtcr_clones.tsv'
    make_10x_clones_file(
        donor_contigs_file,
        organism = 'human_gd', # using 'human_gd' for gdTCR
        clones_file = donor_clones_file,
        stringent = True, # (the default) See Note #1 on clonotype filtering
    )
    # read the GEX data and the clonotypes into CoNGA
    adata = conga.preprocess.read_dataset(
        gex_dir, '10x_mtx', donor_clones_file,
        allow_missing_kpca_file=True)
    disease = donor[:-1]
    adata.obs['disease'] = disease
    adata.obs['disease_int'] = diseases.index(disease) # conga batch ids are integers
    adata.obs['donor'] = donor
    adata.obs['donor_int'] = donor_num # conga batch ids are integers
    all_data.append( adata )

# concatenate the datasets
new_adata = all_data[0].concatenate(all_data[1:])

#save the aggregate AnnData object
new_adata.write('merged_gex_gdtcr.h5ad')
```

References

1. Yost KE, Satpathy AT, Wells DK, Qi Y, Wang C, Kageyama R et al (2019) Clonal replacement of tumor-specific T cells following PD-1 blockade. Nat Med 25(8):1251–1259
2. Wu TD, Madireddi S, de Almeida PE, Banchereau R, Chen Y-JJ, Chitre AS et al (2020) Peripheral T cell expansion predicts tumour infiltration and clinical response. Nature [Internet]. Available from: https://doi.org/10.1038/s41586-020-2056-8
3. Guo X, Zhang Y, Zheng L, Zheng C, Song J, Zhang Q et al (2018) Global characterization of T cells in non-small-cell lung cancer by single-cell sequencing. Nat Med 24(7):978–985
4. Jokinen E, Huuhtanen J, Mustjoki S, Heinonen M, Lähdesmäki H (2019) Determining epitope specificity of T cell receptors with TCRGP [Internet]. bioRxiv. 2019 [cited 2019 Sep 24]. 542332. Available from: https://www.biorxiv.org/content/10.1101/542332v2
5. Zheng C, Zheng L, Yoo J-K, Guo H, Zhang Y, Guo X et al (2017) Landscape of infiltrating T cells in liver cancer revealed by single-cell sequencing. Cell. 169(7):1342–56.e16
6. Zhang L, Yu X, Zheng L, Zhang Y, Li Y, Fang Q et al (2018) Lineage tracking reveals dynamic relationships of T cells in colorectal cancer. Nature 564(7735):268–272
7. Gueguen P, Metoikidou C, Dupic T, Lawand M, Goudot C, Baulande S et al (2021) Contribution of resident and circulating precursors to tumor-infiltrating CD8+ T cell populations in lung cancer. Sci Immunol [Internet]. 6(55). Available from: https://doi.org/10.1126/sciimmunol.abd5778
8. Azizi E, Carr AJ, Plitas G, Cornish AE, Konopacki C, Prabhakaran S et al (2018) Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. Cell. 174(5):1293–308.e36
9. Minervina AA, Pogorelyy MV, Komech EA, Karnaukhov VK, Bacher P, Rosati E et al

(2019) Comprehensive analysis of antiviral adaptive immunity formation and reactivation down to single-cell level [Internet]. bioRxiv. 2019 [cited 2019 Nov 5]. 820134. Available from: https://www.biorxiv.org/content/10.1101/820134v1

10. Zemmour D, Zilionis R, Kiner E, Klein AM, Mathis D, Benoist C (2018) Single-cell gene expression reveals a landscape of regulatory T cell phenotypes shaped by the TCR. Nat Immunol 19(3):291–301
11. Schattgen SA, Guion K, Crawford JC, Souquette A, Barrio AM, Stubbington MJT et al (2021) Integrating T cell receptor sequences and transcriptional profiles by clonotype neighbor graph analysis (CoNGA). Nat Biotechnol [Internet]. Available from: https://doi.org/10.1038/s41587-021-00989-2
12. Luoma AM, Suo S, Williams HL, Sharova T, Sullivan K, Manos M et al (2020) Molecular pathways of colon inflammation induced by cancer immunotherapy. Cell. 182(3):655–671
13. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J et al (2016) Jupyter notebooks - a publishing format for reproducible computational workflows. Positioning Power Acade Publ Players Agents Agendas:87–90
14. Wolf FA, Angerer P, Theis FJ (2018) SCANPY: large-scale single-cell gene expression data analysis. Genome Biol 19(1):15
15. McInnes L, Healy J, Melville J (2018) UMAP: uniform manifold approximation and projection for dimension reduction [Internet]. arXiv [stat.ML]. Available from: http://arxiv.org/abs/1802.03426
16. Shugay M, Bagaev DV, Zvyagin IV, Vroomans RM, Crawford JC, Dolton G et al (2018) VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. Nucleic Acids Res 46(D1):D419–D427
17. Tickotsky N, Sagiv T, Prilusky J, Shifrut E, Friedman N (2017) McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. Bioinformatics 33(18):2924–2929
18. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H et al (2000) The protein data bank. Nucleic Acids Res 28(1):235–242
19. Dash P, Fiore-Gartland AJ, Hertz T, Wang GC, Sharma S, Souquette A et al (2017) Quantifiable predictive features define epitope-specific T cell receptor repertoires. Nature 547(7661):89–93
20. 10x_Genomics. A new way of exploring immunity: linking highly multiplexed antigen recognition to immune repertoire and phenotype (Application Note LIT000047 Rev C) [Internet] (2020). Available from: Retrieved from the 10× Genomics website: https://pages.10xgenomics.com/rs/446-PBO-704/images/10x_AN047_IP_A_New_Way_of_Exploring_Immunity_Digital.pdf
21. Zhang S-Q, Ma K-Y, Schonnesen AA, Zhang M, He C, Sun E et al (2018) High-throughput determination of the antigen specificities of T cell receptors in single cells. Nat Biotechnol [Internet]. Available from: https://doi.org/10.1038/nbt.4282
22. Schneider TD, Stephens RM (1990) Sequence logos: a new way to display consensus sequences. Nucleic Acids Res 18(20):6097–6100
23. DeTomaso D, Yosef N (2021) Hotspot identifies informative gene modules across modalities of single-cell genomics. Cell Syst. 12(5):446–56.e9
24. McKinney W (2010) Data structures for statistical computing in Python. In: Proceedings of the 9th Python in Science Conference [Internet]. SciPy. Available from: https://conference.scipy.org/proceedings/scipy2010/mckinney.html
25. Carter JA, Preall JB, Grigaityte K, Goldfless SJ, Jeffery E, Briggs AW et al (2019) Single T cell sequencing demonstrates the functional role of αβ TCR pairing in cell lineage and antigen specificity. Front Immunol 10:1516
26. Klarenbeek PL, Doorenspleet ME, Esveldt REE, van Schaik BDC, Lardy N, van Kampen AHC et al (2015) Somatic variation of T-cell receptor genes strongly associate with HLA class restriction. PLoS One 10(10):e0140815
27. Emerson R, Sherwood A, Desmarais C, Malhotra S, Phippard D, Robins H (2013) Estimating the ratio of CD4+ to CD8+ T cells using high-throughput sequence data. J Immunol Methods 391(1–2):14–21
28. Li HM, Hiroi T, Zhang Y, Shi A, Chen G, De S et al (2016) TCRβ repertoire of CD4+ and CD8+ T cells is distinct in richness, distribution, and CDR3 amino acid composition. J Leukoc Biol 99(3):505–513