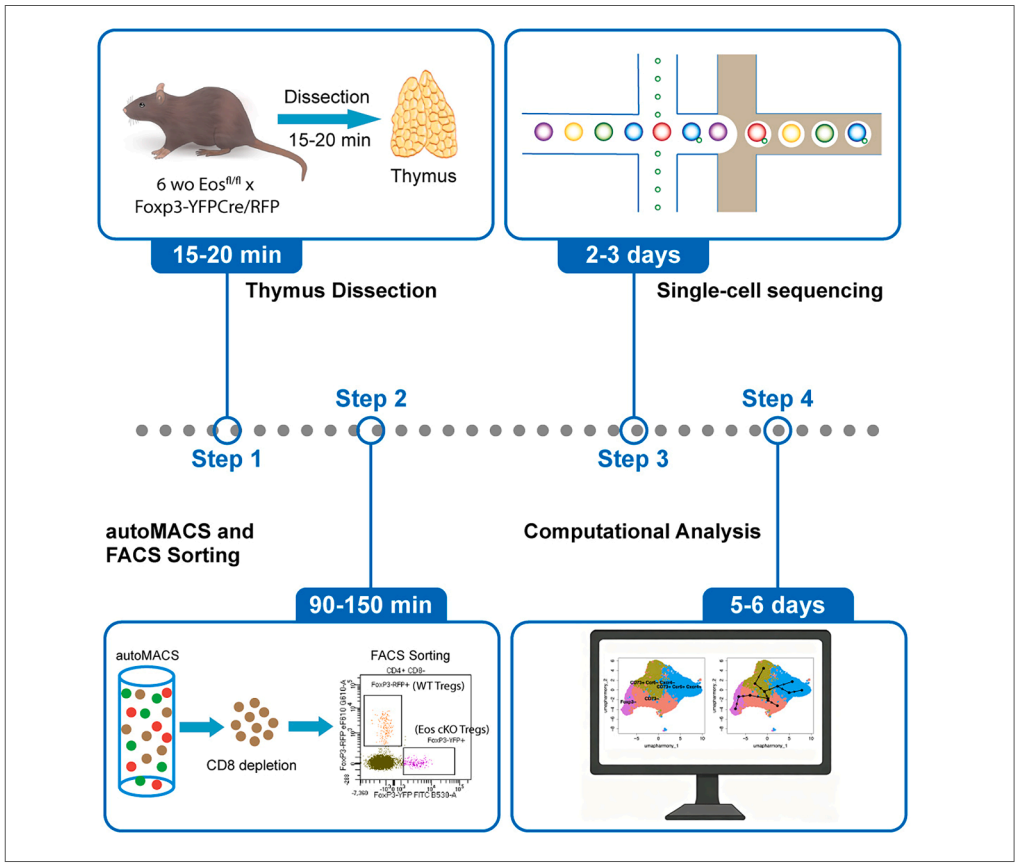
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for identifying recirculating thymic regulatory T cells and characterizing the role of Eos in their function using scRNA-seq and TCR-seq



Previous studies on the characterization of regulatory T cells (Tregs) in the thymus have proven to be simplistic. Here, we present a protocol for identifying recirculating thymic Tregs in mice and characterizing the role of Eos in their function. We describe steps for thymic dissection, autoMACS and fluorescence-activated cell sorting (FACS), single-cell sequencing, and downstream computational analysis. This protocol details a refined classification based on analysis of gene expression from single-cell RNA sequencing (scRNA-seq) and TCR sequencing data.

Xuan Xie, Tom Hill, Francisco Otaizo-Carrasquero, Xunyong Zhou, Ethan M. Shevach

xxsabrina0408@outlook.com (X.X.)
eshevach@niaid.nih.gov (E.M.S.)

Highlights

Isolation and processing workflow of thymic Tregs for single-cell sequencing

Guidance to discriminate Treg subsets using paired scRNA-seq and TCR-seq data

Functional analysis of Eos in recirculating thymic Tregs

Xie et al., STAR Protocols 7, 104620
June 19, 2026 Published by Elsevier Inc.
https://doi.org/10.1016/j.xpro.2026.104620

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for identifying recirculating thymic regulatory T cells and characterizing the role of Eos in their function using scRNA-seq and TCR-seq

Xuan Xie,[1,5,6,*] Tom Hill,[2] Francisco Otaizo-Carrasquero,[3] Xunyong Zhou,[4] and Ethan M. Shevach[1,7,*]

[1]Laboratory of Immune System Biology, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Bethesda, MD, USA

[2]Integrated Data Sciences Section, Research Technologies Branch, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Bethesda, MD, USA

[3]Genomic Research Section, Research Technologies Branch, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Bethesda, MD, USA

[4]Bestzyme & Bestlife Group, Shuyang County, Suqian City, Jiangsu, China

[5]Present address: Hunan Jiyue Biotech, Changsha, Hunan, China

[6]Technical contact

[7]Lead contact

*Correspondence: xxsabrina0408@outlook.com (X.X.), eshevach@niaid.nih.gov (E.M.S.)
https://doi.org/10.1016/j.xpro.2026.104620

SUMMARY

Previous studies on the characterization of regulatory T cells (Tregs) in the thymus have proven to be simplistic. Here, we present a protocol for identifying recirculating thymic Tregs in mice and characterizing the role of Eos in their function. We describe steps for thymic dissection, magnetic cell separator (autoMACS) and fluorescence-activated cell sorting (FACS), single-cell sequencing, and downstream computational analysis. This protocol details a refined classification based on analysis of gene expression from single-cell RNA sequencing (scRNA-seq) and TCR sequencing data.

For details on the use of this protocol, refer to Xie et al.[1]

BEFORE YOU BEGIN

Regulatory T cells (Tregs) are pivotal for immune homeostasis and self-tolerance, with thymic Tregs (tTregs) as the primary subset developed in the thymus.[2,3] Recirculating thymic Tregs, a unique tTreg subpopulation, originate in the thymus, migrate to the periphery, and return, forming a circuit that regulates the de novo generation of Tregs in the thymus.[4,5,6] Early studies identified them using Rag-GFP reporter mice and GFP^{-} expression, a method tied to the role of RAG in T cell development.[7] In a separate study, the majority of GFP^{-} thymic Tregs in Rag-GFP mice were further found to express the cell surface marker CD73.[8] However, this overly simplistic definition fails to account for the existence of mature thymic Tregs that remain resident within the thymus and never egress from this primary lymphoid organ, a distinct subset we term thymus-resident Tregs.[5,7–9] With respect to the molecular mechanisms governing their recirculation, accumulating evidence has implicated the Ccr6/Ccl20 and Cxcr4/Cxcl12 chemokine axes in this process.[4,5] In addition to the known role of Eos (Ikzf4) in Treg stability, we have further identified a direct association between Eos and the expression of Ccr6 and Cxcr4 in thymic Tregs.[1,10] To better define sub-populations of thymic Tregs and investigate the role of Eos in recirculating thymic Tregs, we describe a comprehensive approach to characterize heterogeneous thymic Treg populations using single-cell RNA sequencing (scRNA-seq) and T cell receptor sequencing (TCR-seq) profiling of thymic Tregs isolated from 6 weeks-old (wo) $Eos^{fl/fl}$ x Foxp3-YFPCre/RFP female mice (heterozygous female mice). We used $Eos^{fl/fl}$ × Foxp3-YFPCre/RFP

heterozygous female mice because Foxp3 is located on the X chromosome. Random X-chromosome inactivation yields ~50% Cre-expressing Eos cKO (conditional knockout) and ~50% Cre-negative wild-type (WT) Tregs in the same mouse, providing robust internal controls for studying Eos function. Due to the survival disadvantage of Eos cKO Tregs, we selected 6 wo heterozygous female mice for this study to ensure comparable numbers of Eos cKO and WT Tregs.[1] Concomitantly, this approach enables investigation into the functional role of Eos in recirculating thymic Tregs. Notably, the protocol is readily adaptable for the analysis of scRNA-seq and TCR-seq data from wild-type (WT) mice when Eos-mediated functions are not the focus of investigation, and can be further modified to characterize Treg subpopulations isolated from extrathymic tissues and extended for application to human cells.

Innovation

This protocol advances the study of thymic Tregs by overcoming the limitations of traditional, overly simplistic classification. Conventional approaches rely solely on Rag-GFP reporter mice and GFP^- status to define recirculating thymic Tregs, ignoring mature thymus-resident Tregs that never exit the thymus.[5,7–9] Key chemokine axes, including Ccr6/Ccl20 and Cxcr4/Cxcl12, have emerged as critical regulators of thymic Treg trafficking and recirculation, as supported by recent publications, yet these markers have not been comprehensively integrated into high-resolution thymic Treg subset definitions.[4,5,11] In contrast, we establish a refined, multi-omic workflow integrating scRNA-seq and TCR-seq for unbiased subpopulation identification. We optimize thymic Treg isolation from heterozygous female $Eos^{fl/fl}$ × Foxp3-YFPCre/RFP mice, enabling simultaneous characterization of recirculating thymic Tregs and the functional role of the transcription factor Eos. Unlike prior methods, this framework distinguishes bona fide recirculating subsets using gene expression and clonal TCR profiling rather than surrogate reporters. The protocol is modular: adaptable to WT mice for Eos-independent studies and can be extended to Tregs from extrathymic tissues. By integrating experimental isolation and customized computational analysis into a unified pipeline, this work delivers a more accurate, high-resolution strategy for dissecting Treg heterogeneity and recirculation biology.

Institutional permissions (if applicable)

All studies were approved by the NIAID Animal Care and Use Committee (Protocols LISB 15E and LISB 8E) and conducted in compliance with federal and local institutional regulatory standards. Mice were housed in ventilated racks under a standard 12-hour light–dark cycle. All animal experiments should be performed only after obtaining the requisite institutional approvals and completing the required animal care and experimental training.

Preparation for thymus dissection and dissociation

Timing: 15–20 min

1. Disinfect a biosafety hood with 70% ethanol, and turn on the air flow for at least 15 minutes to establish a sterile environment. All surfaces within the hood are wiped dry with sterile gauze to avoid residual ethanol contamination.
2. Prepare sterile forceps, scissors, 6 mL syringes (plungers separated from barrels), cell strainers, sterile Petri dishes, and 50 mL centrifuge tubes in advance and place them in the biosafety hood. Keep Collection Medium and autoMACS Running Buffer at 4 °C (on ice) and verify their sterility before use.
3. Wear sterile gloves, a lab coat, and protective eyewear throughout the preparation and subsequent procedures to prevent cross-contamination and ensure operator safety.

⚠ CRITICAL: It is critical to properly disinfect the biosafety hood with 70% ethanol, run the airflow for at least 15 min, and dry all surfaces to eliminate ethanol residue; maintain Collection Medium and autoMACS Running Buffer at 4 °C while confirming their sterility;

and consistently wear sterile gloves, a lab coat, and protective eyewear to prevent cross-contamination and protect the operator during all procedures.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Antibodies		
Mouse Fc Block (50-fold dilution)	BD Biosciences	Cat#: 553142; RRID: AB_394657
Anti-mouse CD4-PerCP-Cy5.5 (50-fold dilution)	BD Biosciences	Cat#: 569923; RRID: AB_3685412
Anti-mouse CD8-APC (50-fold dilution)	BD Biosciences	Cat#: 553035; RRID: AB_398527
Chemicals, peptides, and recombinant proteins		
autoMACS Running Buffer	Miltenyi Biotec	Cat#: 130-091-221
autoMACS Washing Solution	Miltenyi Biotec	Cat#: 130-092-987
RPMI 1640	Thermo Fisher Scientific	Cat#: 21870092
RPMI 1640 without phenol red	Thermo Fisher Scientific	Cat#: 32404014
FBS	Sigma-Aldrich	Cat#: 23E082
HEPES	Thermo Fisher Scientific	Cat#: 15630130
PBS, pH 7.4	Thermo Fisher Scientific	Cat#: 10010049
Critical commercial assays		
CD8a (Ly-2) MicroBeads, mouse	Miltenyi Biotec	Cat#: 130-117-044
Chromium GEM-X Single Cell 5′ Kit v3 16 rxns	10× Genomics	Cat#: PN-1000699
Library Construction Kit C 16 rxns	10× Genomics	Cat#: PN-1000694
Chromium GEM-X Single Cell 5′ Chip Kit 4 chips	10× Genomics	Cat#: PN-1000698
Mouse 16 rxns TCR	10× Genomics	Cat#: PN-1000254
Dual Index Kit TT Set A 96 rxns	10× Genomics	Cat#: PN-1000215
Deposited data		
Single Cell RNA Sequencing Data	Xie et al.[1]	GEO: GSE305537
Experimental models: Organisms/strains		
$Eos^{fl/fl}$ x Foxp3-YFPCre/RFP; female Mus musculus; Eos-fl/fl x Foxp3-YFPCre/RFP; 6 weeks	Xie et al.[1]	https://doi.org/10.1016/j.celrep.2025.116838
Software and algorithms		
Cellranger mkfastq (Bcl2fastq 2.20)	10× Genomics	https://www.10xgenomics.com/
Cellranger 8.0.1	10× Genomics	https://www.10xgenomics.com/
R v4.4.2	The R Project	http://r-project.org
Seurat 5.4.0	Stuart et al.[12]	https://doi.org/10.1016/j.cell.2019.05.031
dplyr 1.2.0	Wickham et al.[13]	https://dplyr.tidyverse.org/
ggplot2 4.0.2	Wickham[14]	https://ggplot2.tidyverse.org
scRepertoire 2.2.1	Borcherding et al.[15]	https://doi.org/10.12688/f1000research.22139.2
djvdj 0.1.0	Sheridan[16]	https://rnabioco.github.io/djvdj/
stringr 1.6.0	Wickham[17]	https://stringr.tidyverse.org
harmony 1.2.4	Korsunsky et al.[18]	https://doi.org/10.1038/s41592-019-0619-0
celldex 1.20.0	Bioconductor[19]	https://bioconductor.org/packages/celldex/
SingleR 2.12.0	Bioconductor[19]	https://bioconductor.org/packages/SingleR/
ggrepel 0.9.8	Slowikowski[20]	https://ggrepel.slowkow.com/
readxl 1.4.5	Wickham et al.[21]	https://readxl.tidyverse.org
ggpubr 0.6.3	Kassambara et al.[22]	https://rpkgs.datanovia.com/ggpubr/
viridis 0.6.5	Garnier et al.[23]	https://sjmgarnier.github.io/viridis/
slingshot 2.18.0	Street et al.[24]	https://doi.org/10.1186/s12864-018-4772-0
tradeSeq 1.24.0	Van den Berge et al.[25]	https://doi.org/10.1038/s41467-020-14766-3
RColorBrewer v1.1-3	The R Project	https://CRAN.R-project.org/package=RColorBrewer
abdiv v0.2.0	The R Project	https://CRAN.R-project.org/package=abdiv
shape 1.4.6.0	The R Project	https://CRAN.R-project.org/package=shape
Adobe Illustrator (25.0.1)	Adobe	https://www.adobe.com/products/illustrator

(*Continued on next page*)

Continued		
REAGENT or RESOURCE	SOURCE	IDENTIFIER
Other		
NovaSeq X Plus	Illumina	N/A
autoMACS Pro Separator	Miltenyi Biotec	N/A
FACSAria IIIu Sorter	BD Biosciences	N/A

MATERIALS AND EQUIPMENT

Collection Medium		
Reagent	Final concentration	Amount
RPMI 1640	90%	450 mL
FBS	10%	50 mL
Total	N/A	500 mL
Store at 4 °C for up to 3 months.		

⚠ CRITICAL: In this collection medium formulation, RPMI 1640 is non-toxic with only mild irritant potential, while FBS is the main biohazardous reagent requiring BSL-2 containment and standard bloodborne pathogen precautions. Prepare the buffer in a biological safety cabinet to maintain sterility. All manipulations should follow standard laboratory safety protocols to avoid biological and chemical exposure.

Alternatives: RPMI 1640 can be replace by DMEM, MEM and IMDM.

FACS Sorting Buffer		
Reagent	Final concentration	Amount
RPMI 1640 without phenol red	89%	445 mL
FBS	10%	50 mL
HEPES	1%	5 mL
Total	N/A	500 mL
Store at 4 °C for up to 3 months.		

⚠ CRITICAL: In this collection media formulation, FBS may carry potential infectious agents, while HEPES can irritate the eyes, skin, and respiratory tract, and RPMI 1640 medium exhibits no significant acute toxicity under normal laboratory handling. RPMI 1640 medium without phenol red is a critical component for preparing FACS Sorting Buffer, because phenol red can interfere with fluorescence signals, increase background fluorescence, and reduce the purity and accuracy of cell sorting. Prepare the buffer in a biological safety cabinet to maintain sterility for cell sorting. All manipulations should follow standard laboratory safety protocols to avoid biological and chemical exposure.

STEP-BY-STEP METHOD DETAILS

Thymus dissection and dissociation

Timing: 35–50 min

This section describes the procedure for dissociating the thymus into single-cell suspensions for use in subsequent experiments. Following surgical isolation, the thymus should be immediately

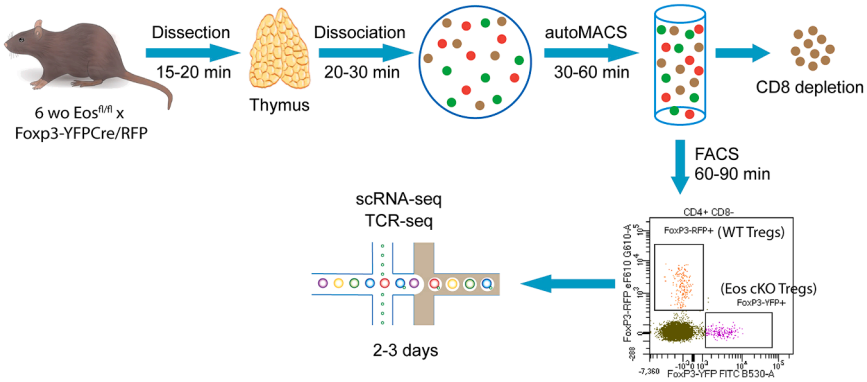


Figure 1. Experimental workflow and single-cell analysis of thymic Tregs from heterozygous female mice

Schematic of the experimental strategy: 6 wo Eos$^{fl/fl}$ x Foxp3-YFPCre/RFP mice were used for thymus dissection (15–20 min), followed by tissue dissociation (20–30 min).

CD8$^+$ cells were depleted via autoMACS (30–60 min), and Foxp3-RFP$^+$ (WT Tregs) and Foxp3-YFP$^+$ (Eos cKO Tregs) cells were sorted by FACS (60–90 min) for scRNA-seq and TCR-seq (2–3 days).

transferred into a sterile Petri dish containing collection medium and a cell strainer, and maintained on ice prior to processing. To preserve cellular viability, thymic tissue should be processed as soon as possible following collection (Figure 1).

1. Euthanize mice in strict accordance with the institutional animal care and use committee (IACUC) protocols and relevant policies.
2. Carefully excise the thymus using sterile forceps and scissors.
3. Place the thymus in a cell strainer, and subsequently maintain the strainer immersed in Collection Medium.
4. Release cells from the thymic tissue by gently grinding the organ using the rubber tip of a 6 mL syringe plunger (pre-remove the plunger from the syringe barrel first).
5. Transfer the cell strainer to a new 50 mL centrifuge tube, and transfer the cell suspension from the petri dish to the same 50 mL tube.
6. Rinse the collection dish with 2 mL collection medium, and transfer the rinsing medium to the 50 mL tube through the cell strainer.
7. Pellet the cells by centrifugation (~200 × *g*, 5min), then decant the supernatant.
8. Resuspend the cell pellet in 5 mL autoMACS Running Buffer, and transfer the resulting cell suspension into a new tube through a cell strainer to further remove clumps.

⚠ CRITICAL: All steps must be performed using sterile techniques (wear sterile gloves, use autoclaved instruments/pre-sterilized reagents, work in a 70% ethanol-disinfected biosafety hood) to avoid contamination, which is crucial for cell viability and experimental validity. Strict adherence to IACUC protocols is mandatory for ethical compliance.

Note: A minimum thymocyte yield of ≥130 million cells is expected to ensure sufficient numbers of thymic Tregs for downstream analyses. If the actual cell yield is lower than expected, potential solutions are provided in Troubleshooting 1.

CD8$^+$ T cell depletion by autoMACS

Timing: 30–60 min

Due to the large numbers of CD4$^+$CD8$^+$ double-positive (DP) thymocytes in the thymus, use the autoMACS system to deplete CD8$^+$ T cells from thymocytes in order to enrich CD4$^+$ T cells.

9. Spin down the cells (~200 × *g*, 5min) and resuspend cell pellet in 90 μL of autoMACS Running Buffer per 10^7 total cells.
10. Add 10 μL of CD8a (Ly-2) MicroBeads per 10^7 total cells.
11. Mix well and incubate for 10min in the refrigerator (4 °C).
12. Run the autoMACS using the Depletes program and collect the negative fraction.

⚠ CRITICAL: Maintain all cells and reagents at 4 °C throughout the procedure to preserve cell viability. This step can be skipped if analyzing samples derived from tissues with low $CD8^+$ T cell abundance.

FACS for scRNA-seq

Timing: 60–90 min

This section describes the sorting of WT (Foxp3-RFP^+) and Eos cKO (Foxp3-YFP^+) thymic Tregs from 6 wo $Eos^{fl/fl}$ x Foxp3-YFPCre/RFP heterozygous female mice. In 6 wo heterozygous female mice, there is a similar ratio of WT and Eos cKO thymic Tregs due to X-chromosome inactivation. Using 6 wo mice guarantees a sufficient cell number of Eos cKO Tregs, which have a survival disadvantage *in vivo*.[1]

13. Centrifuge the cells from the negative fraction (~200 × *g*, 5min) and resuspend the cell pellet in 1000 μL FACS Sorting Buffer.
14. Add 20 μL of Mouse Fc Block and incubate with the cells for 10 min at RT.
15. Add 20 μL of anti-mouse CD4-PerCP-Cy5.5 and anti-mouse CD8-APC antibodies to the cells and incubate for 12–16 h at 4 °C.
16. Wash the cells with FACS Sorting Buffer and resuspend the cell pellet in 1000 μL FACS Sorting Buffer.
17. Strain the cell suspension into a new FACS tube using a 70 μm cell strainer.
18. Sort desired populations into FACS Sorting Buffer in sterile, capped tubes and transfer onto ice.

Note: See Figure 2 for gating strategy.

19. Wash cells with PBS supplemented with 10% FBS, centrifuge (~200 × *g*, 5min) and then aspirate the supernatant to leave behind a volume of ~50 μL.
20. Add PBS supplemented with 10% FBS to have 1300-1600 cells/μL. Each sample should be at least 50 μL. Using a regular-bore pipette tip, resuspend the cell pellet in the leftover supernatant by gently pipetting up and down 10–15 times.
21. If necessary, adjust the volume to obtain the target cell concentration.
22. Once the target cell concentration is obtained, place the cells on ice.

Single-cell sequencing

Timing: 2–3 days

Here, we outline the procedures for performing single-cell sequencing using Chromium GEM-X Single Cell 5′ Reagent Kits v3 in accordance with the official manufacturer's specifications provided by 10x Genomics.

23. Based on the 10x Genomics Cell Suspension Volume Calculator Table, add the appropriate amount of cell suspension and PBS to the Master Mix.
24. Load the GEM-X Chip with Master Mix + Cell Suspension, Gel Beads, and Partitioning Oil B as specified, and assemble the chip with the Chromium X/iX Chip Holder.

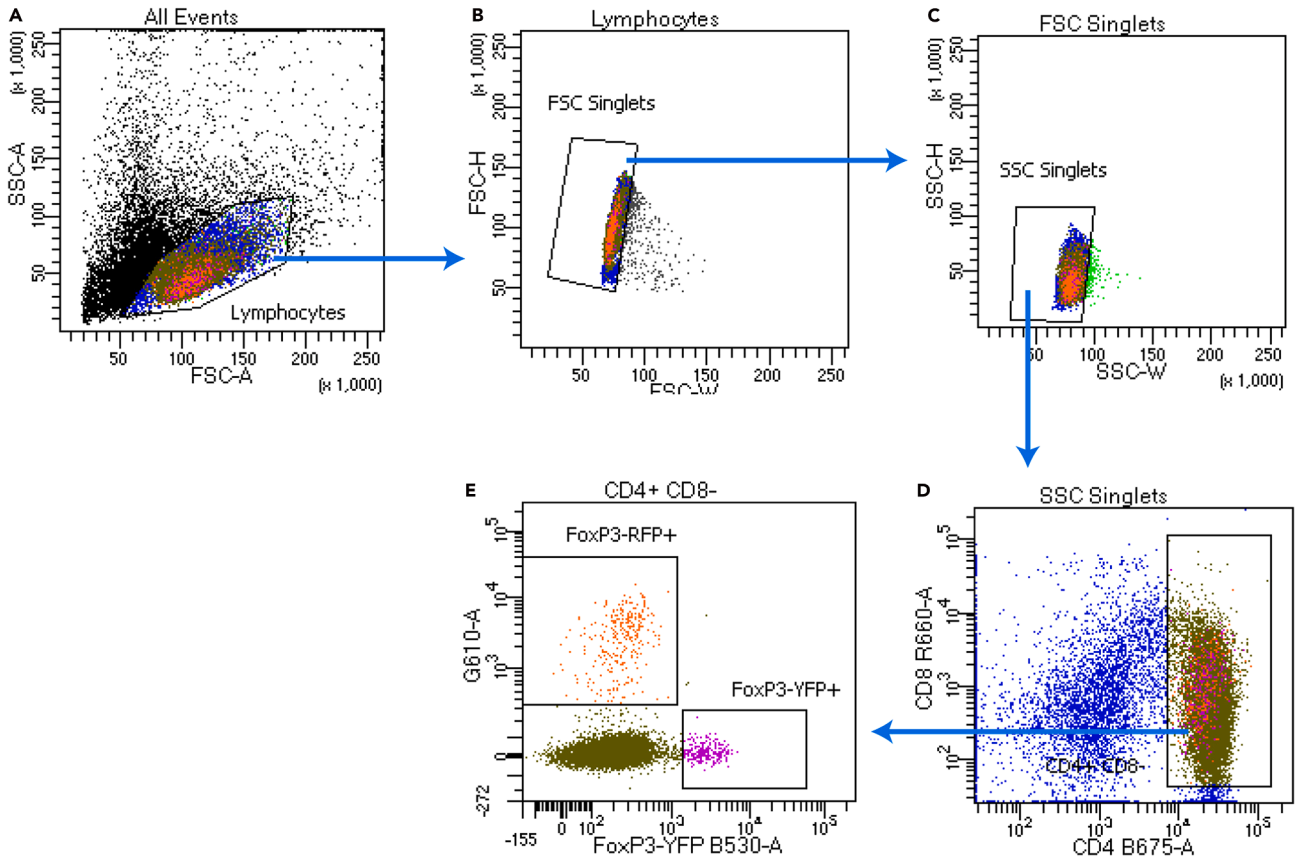


Figure 2. Flow cytometry gating strategy for the isolation of thymic Tregs

Representative flow cytometry plots showing the sequential gating strategy used to purify thymic Tregs.

(A) All Events: Initial plot of forward scatter area (FSC-A) vs. side scatter area (SSC-A) to identify the lymphocyte population.

(B) Lymphocytes: Gated on the lymphocyte population and plotted as FSC-A vs. FSC-H to select FSC singlets, excluding doublets.

(C) FSC Singlets: FSC singlets are further analyzed by SSC-H vs. SSC-W to gate on SSC singlets, ensuring single-cell analysis.

(D) SSC Singlets: SSC singlets are stained for CD4 and CD8, and gated to select CD4+ CD8– thymocytes.

(E) CD4+ CD8–: CD4+ CD8– thymocytes are analyzed for Foxp3-YFP and Foxp3-RFP expression to identify Foxp3-RFP+ (WT Tregs) and Foxp3-YFP+ (Eos cKO Tregs) populations for subsequent experiments.

25. Insert the assembled chip into the Chromium X/iX (firmware ≥2.0.0) and run the instrument to generate GEMs. Troubleshooting 2.
26. After GEM generation, carefully transfer the GEMs to an 8-strip tube and complete the GEM-RT incubation as specified in the protocol.
27. Proceed with the 10x Genomics protocol including Post GEM-RT Cleanup, cDNA Amplification, cDNA Cleanup, and Post cDNA Amplification QC & Quantification.
28. For V(D)J sequencing: Perform V(D)J Amplification (1&2), corresponding cleanups, and Post V(D)J Amplification QC & Quantification.
29. For 5′ Gene Expression sequencing: Take 25% of the amplified cDNA and perform GEX Fragmentation, End Repair & A-tailing, cleanups, and subsequent steps.
30. Construct V(D)J and/or 5′ Gene Expression libraries via Adaptor Ligation, Sample Index PCR, and Post PCR Cleanup.
31. After library construction, run QC using an Agilent Bioanalyzer High Sensitivity chip (1:10 dilution for libraries). Troubleshooting 3.
32. Submit pooled libraries for sequencing based on 10x Genomics recommended run parameters (paired-end, dual indexing). Troubleshooting 4.

Samples Data Statistics

A

Sample	Gene Expression Cells	Gene Expression Mean reads per cell	Gene Expression Median genes per cell	Gene Expression Total genes detected	Gene Expression Median UMI counts per cell
F1_Ikzf4_cKO_thymic_Tregs	16,705	47,016	3,249	22,434	10,128
F1_WT_thymic_Tregs	14,798	51,761	3,208	22,212	9,705
F2_Ikzf4_cKO_thymic_Tregs	18,359	40,446	3,177	22,539	9,796
F2_WT_thymic_Tregs	18,190	41,060	3,144	22,497	9,393

B

Sample	VDJ T Estimated number of cells	VDJ T Mean reads per cell	VDJ T Mean used reads per cell	VDJ T Number of cells with productive V-J spanning pair	VDJ T Paired clonotype diversity
F1_Ikzf4_cKO_thymic_Tregs	14,394	12,210	9,953	13,118	11,541
F1_WT_thymic_Tregs	12,466	11,661	9,297	11,291	7,599
F2_Ikzf4_cKO_thymic_Tregs	15,685	10,392	8,381	14,188	13,805
F2_WT_thymic_Tregs	15,032	11,400	9,015	13,532	9,877

Figure 3. Summary of single-cell RNA-seq and TCR-seq data statistics for thymic Treg cells from Ikzf4 (Eos) cKO and WT mice

(A) Top table: Gene expression statistics for each sample.

(B) Bottom table: V(D)J TCR sequencing statistics for each sample.

Note: See Figure 3 for sample statistics.

Single-cell analysis

Timing: 5–6 days

Here, we describe the computational workflow for the analysis of scRNA-seq/VDJ data of thymic Tregs from 6 wo heterozygous female mice, which involves quality control (QC), integration, annotation, DE analysis, TCR repertoire profiling, trajectory inference, and visualization (Figure 4). The total computation time varies depending on the hardware specifications of the system used. Single-cell RNA-seq analysis was performed using 100 GB RAM and 4 CPU cores; a minimum of 50 GB RAM and 4 CPU cores is sufficient for comparable datasets.

33. Using the code below or in supplemental information, load the necessary libraries into R.

```
library(Seurat)

library(dplyr)

library(ggplot2)

library(scRepertoire)

library(djvdj)

library(stringr)

library(harmony)
```

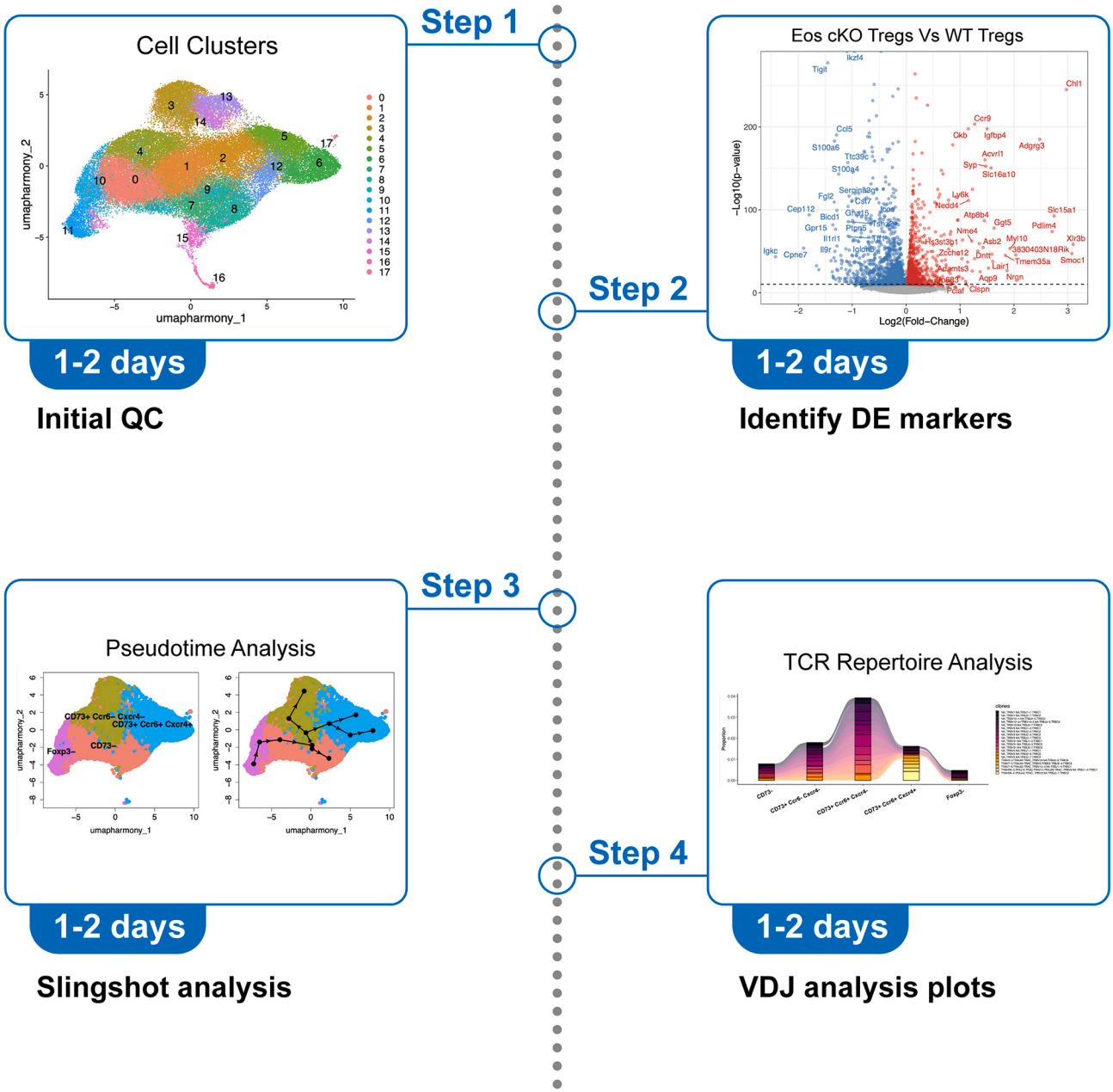


Figure 4. Integrated workflow for single-cell RNA-seq and TCR repertoire analysis of Treg cells

This figure outlines a multi-step analytical pipeline for dissecting the molecular and clonal dynamics of thymic Tregs. We have reused parts of figures from our previously published research articles (Xie et al., 2025).[1] All necessary permissions have been obtained.

```
library(celldex)

library(SingleR)

library(RColorBrewer)

library(abdiv)
```

```
library(ggrepel)

library(readxl)

library(ggpubr)

library(viridis)

library(slingshot)

library(tradeSeq)

library(shape)
```

34. Download the example data from GEO: GSE305537.[1]
35. Define base path and sample directories.

```
path <- ~/Project_Path
```

Note: Replace path with the real directory containing the actual 10x Genomics output.

36. Load gene expression data into Seurat.
    a. Get gene expression directories.
    b. Load gene expression data into Seurat.

```
# Gene Expression (GEX) directories

gex_dirs <- c(

F1_Ikzf4_cKO_thymic_Tregs   =   file.path(path,"F1_Ikzf4_cKO_thymic_Tregs/
count/sample_filtered_feature_bc_matrix"),

F1_WT_thymic_Tregs     = file.path(path, "F1_WT_thymic_Tregs/count/sample_
filtered_feature_bc_matrix"),

F2_Ikzf4_cKO_thymic_Tregs   =   file.path(path,"F2_Ikzf4_cKO_thymic_Tregs/
count/sample_filtered_feature_bc_matrix"),

F2_WT_thymic_Tregs     = file.path(path, "F2_WT_thymic_Tregs/count/sample_
filtered_feature_bc_matrix")

)

# Load GEX data into Seurat

so <- gex_dirs |>

  Read10X() |>

  CreateSeuratObject()
```

37. Integrate VDJ data using djvdj.
    a. Get VDJ data directories.

b. Integrate VDJ data using djvdj.

```
#VDJ directories

vdj_dirs <- c(

  F1_Ikzf4_cKO_thymic_Tregs = file.path(path, "F1_Ikzf4_cKO_thymic_Tregs/
vdj_t/"),

  F1_WT_thymic_Tregs = file.path(path, "F1_WT_thymic_Tregs/vdj_t/"),

  F2_Ikzf4_cKO_thymic_Tregs = file.path(path, "F2_Ikzf4_cKO_thymic_Tregs/
vdj_t/"),

  F2_WT_thymic_Tregs = file.path(path, "F2_WT_thymic_Tregs/vdj_t/")

)

# Integrate VDJ data using djvdj

so <- so |>

  import_vdj(

    vdj_dirs,

    define_clonotypes = "cdr3_gene",

    include_mutations = TRUE

  )
```

38. Quality control (QC) and Metadate Processing (Figures 5A and 5B).
    a. Calculate mitochondrial percentage.
    b. Extract metadata information from cell IDs (e.g., replicate and genotype).
    c. Update Seurat metadata.
    d. Save unfiltered object.
    e. QC Visualizations.

```
# Calculate mitochondrial percentage

so[["percent.mt"]] <- PercentageFeatureSet(so, pattern = "^mt-")

# Extract metadata information from cell IDs (e.g., replicate and genotype)

meta_info <- str_split_fixed(row.names(so@meta.data), "_", 6)

mutant_label <- meta_info[, 2]

mutant_label[mutant_label == "Ikzf4"] <- "Ikzf4 cKO"

# Update Seurat metadata
```

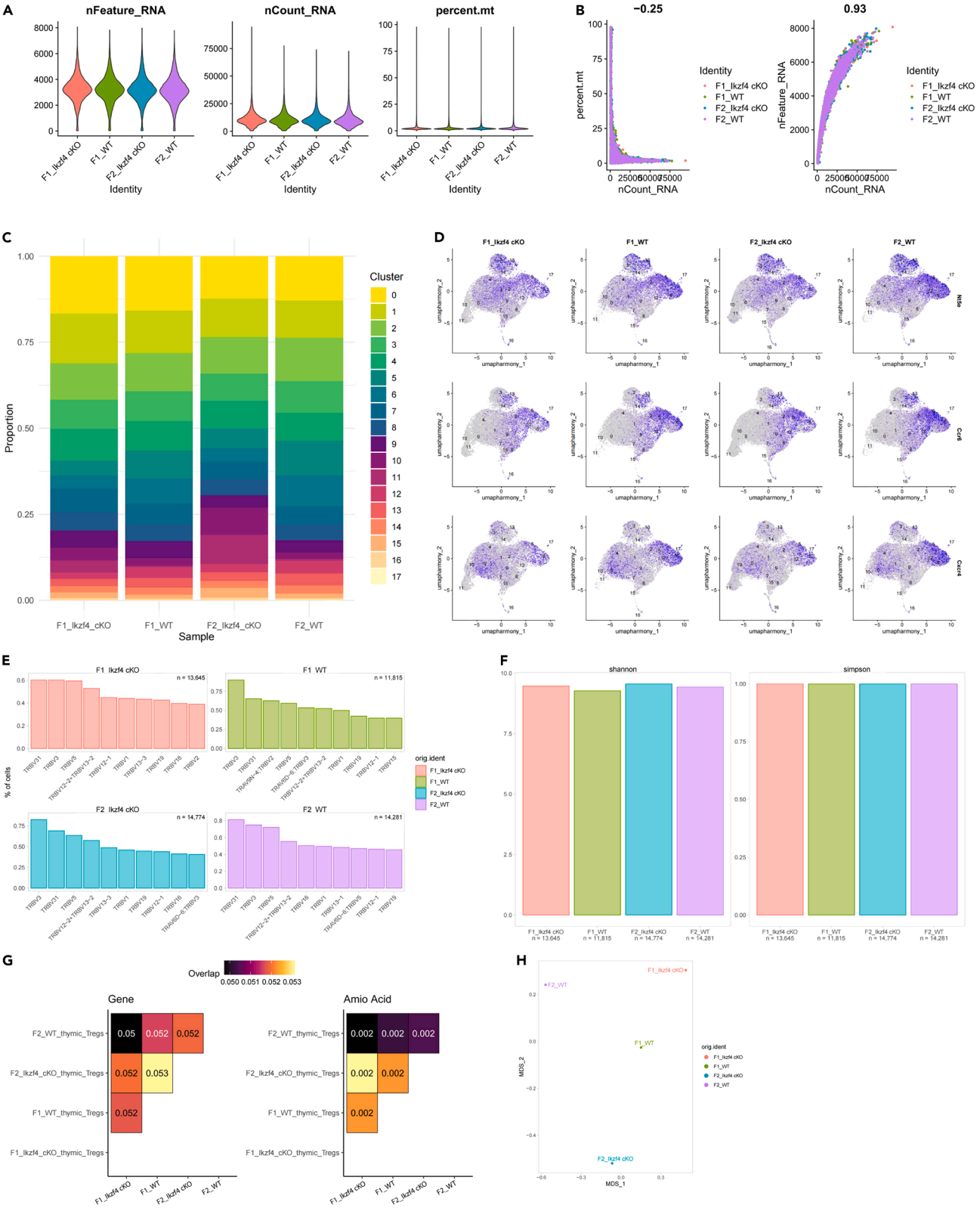

A
nFeature_RNA
nCount_RNA
percent.mt
F1_Ikzf4 cKO
F1_WT
F2_Ikzf4 cKO
F2_WT
Identity
B
−0.25
0.93
percent.mt
nFeature_RNA
nCount_RNA
C
Cluster
Proportion
Sample
D
F1_Ikzf4 cKO
F1_WT
F2_Ikzf4 cKO
F2_WT
umapharmony_1
umapharmony_2
E
% of cells
orig.ident
F1_Ikzf4 cKO
F1_WT
F2_Ikzf4 cKO
F2_WT
F
shannon
simpson
G
Overlap
Gene
Amio Acid
F2_WT_thymic_Tregs
F2_Ikzf4_cKO_thymic_Tregs
F1_WT_thymic_Tregs
F1_Ikzf4_cKO_thymic_Tregs
H
MDS_1
MDS_2

Figure 5. Key quality control, cell heterogeneity and T cell clonal profiling of thymic Treg single-cell RNA and TCR sequencing datasets

(A) Violin plots showing distributions of single-cell RNA-seq quality control metrics across thymic Treg samples, including detected gene counts (nFeature_RNA), transcript counts (nCount_RNA), and percentage of mitochondrial reads (percent.mt).

(B) Correlation scatter plots assessing relationships between transcript counts and mitochondrial read ratio (left), and transcript counts versus detected gene features (right), stratified by sample genotype and replicate identity.

(C) Cluster composition across samples. Stacked bar plot showing the proportion of cells in each Seurat cluster for all four samples.

(D) UMAP feature plots displaying normalized expression levels of key Treg signature and functional genes (Nt5e, Ccr6, Cxcr4) across clusters and samples.

(E) Frequency distributions of TCR V-gene usage across experimental sample groups.

(F) TCR repertoire diversity analysis showing Shannon and Simpson diversity indices across samples.

(G) Circos plot illustrating clonotype overlap and similarity between WT and Ikzf4 cKO groups.

(H) Multidimensional scaling (MDS) visualization of TCR clonotype relatedness across samples.

```
so$mutant <- mutant_label

so$replicate <- so$orig.ident

so$orig.ident <- paste(so$replicate, so$mutant, sep = "_")

# Save unfiltered object

saveRDS(so, file.path(path, "merged.unfiltered.rds"))

# QC Visualizations

v1 <- VlnPlot(so, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"),

       ncol = 3, group.by = "orig.ident", pt.size = 0)

plot1 <- FeatureScatter(so, feature1 = "nCount_RNA", feature2 = "percent.mt",
group.by = "orig.ident")

plot2 <- FeatureScatter(so, feature1 = "nCount_RNA", feature2 = "nFeature_
RNA", group.by = "orig.ident")

ggsave(file.path(path, "raw_violin.pdf"), plot = v1, width = 10, height = 4)

ggsave(file.path(path, "raw_scatter.pdf"), plot = plot1 + plot2, width = 8,
height = 4)
```

39. Filtering and Initial Processing.
    a. Filter low quality cells.
    b. Split layers for integration (Seurat v5 style).
    c. Transforms raw data into a normalized, scaled, and dimensionally reduced format.
    d. Run UMAP across multiple dimensions for comparison.

```
# Filter low quality cells

so <- subset(so, subset = nFeature_RNA > 200 & nFeature_RNA < 6000 &

       percent.mt < 10 & nCount_RNA < 25000)

# Split layers for integration (Seurat v5 style)
```

```
so@assays$RNA <- split(so@assays$RNA, f = so$orig.ident)

# Standard pipeline

so <- NormalizeData(so)

so <- FindVariableFeatures(so, selection.method = "vst", nfeatures = 2000)

so <- ScaleData(so)

so <- RunPCA(so, features = VariableFeatures(so))

# Run UMAP across multiple dimensions for comparison

so <- FindNeighbors(so, dims = 1:10)

so <- FindClusters(so, resolution = 0.25)

for (d in c(510, , 2030, , 40)) {

  so <- RunUMAP(so, dims = 1:d, reduction.name = paste0("umap_", d), verbose =
FALSE)

}

saveRDS(so, file.path(path, "merged.filtered.rds"))
```

40. Integrate using Harmony and CCA methods.
    a. Data integration.
    b. Process CCA Integration Results.
    c. Process Harmony integration results.

```
# Integrate using Harmony and CCA methods

so <- IntegrateLayers(so, method = HarmonyIntegration, orig.reduction = "pca",

          new.reduction = 'harmony', verbose = FALSE)

so <- IntegrateLayers(so, method = CCAIntegration, orig.reduction = "pca",

          new.reduction = 'cca', verbose = FALSE)

# Process CCA integration results

so <- FindNeighbors(so, reduction = "cca", dims = 1:20)

so <- RunUMAP(so, reduction = "cca", dims = 1:20, reduction.name = "umap.cca")

so <- FindClusters(so, resolution = 0.5, cluster.name = "cca_clusters")

# Process Harmony integration results

so <- FindNeighbors(so, reduction = "harmony", dims = 1:20)
```

```
so <- RunUMAP(so, reduction = "harmony", dims = 1:20, reduction.name =
"umap.harmony")

so <- FindClusters(so, resolution = 0.5, cluster.name = "harmony_clusters")
```

Note: We selected Harmony for its strong performance in batch correction while preserving biologically meaningful cellular heterogeneity, making it ideal for integrating multiple samples in our single-cell dataset.[18] We also included CCA as a complementary integration approach due to its robust cross-sample alignment and widespread use in standard Seurat-based workflows, ensuring our integration strategy is reliable, well-validated, and comparable to field standards.[26]

41. Cell Type Annotation using SingleR. Troubleshooting 5.
    a. Preparing the Seurat object for annotation (joining layers).
    b. Using the ImmGen reference dataset (a gold-standard for immune cell gene expression) to predict broad ("main") and detailed ("fine") cell type labels for every cell.
    c. Cleaning up messy fine labels (removing extra text in parentheses).

```
so <- JoinLayers(so) # Re-join layers for SingleR

ImmGen <- ImmGenData()

# Main Labels

pred.main <- SingleR(test = GetAssayData(so, assay = "RNA", slot = "data"),

          ref = ImmGen, labels = ImmGen$label.main)

so[["SingleR5"]] <- pred.main$labels

#Fine Labels

pred.fine <- SingleR(test = GetAssayData(so, assay = "RNA", slot = "data"),

          ref = ImmGen, labels = ImmGen$label.fine)

so[["SingleR6"]] <- pred.fine$labels

# Clean up fine labels (remove text in parentheses)

so[["SingleR7"]] <- str_split_fixed(as.vector(so$SingleR6), " \\(", 2)[, 1]

saveRDS(so, file.path(path, "int.filtered.rds"))
```

Note: ImmGen database was employed as the reference for SingleR annotation, as it is the gold-standard reference dataset for mouse immune cell transcriptomes, ensuring reliable cell type identification.[27]

42. Narrow down the full dataset to only T cells.
    a. Recalculate cell neighbors/clusters (tailored to T cell-specific variation).
    b. Re-process the T cell subset.
    c. Save the T cell-specific Seurat object for downstream analysis.

```
# Broad T cell selection based on fine labels

t_cell_types <- c("T cells (T.CD4TESTCJ)", "T cells (T.Tregs)", "T
cells (T.4.Pa)",

         "T cells (T.4.PLN)", "T cells (T.4FP3+25+)", "T cells (T.4Mem)",

         "T cells (T.4MEM44H62L)", "T cells (T.4Nve)", "T cells (T.4NVE)",

         "T cells (T.4SP24-)", "T cells (T.4SP24int)", "T cells (T.4SP69+)",

         "T cells (T.CD4.48H)", "T cells (T.CD4.5H)", "T cells (T.CD4.96H)",

         "T cells (T.CD4CONTROL)", "T cells (T.DP69+)")

so2 <- subset(so, subset = SingleR6 %in% t_cell_types)

# Re-process so2

so2 <- FindNeighbors(so2, reduction = "harmony", dims = 1:20)

so2 <- RunUMAP(so2, reduction = "harmony", dims = 1:20, reduction.name =
"umap.harmony")

so2 <- FindClusters(so2, resolution = 0.5, cluster.name = "harmony_clusters")

so2 <- FindClusters(so2, resolution = 1, cluster.name = "harmony_clusters.1")

saveRDS(so2, file.path(path, "int.filtered2.rds"))
```

43. Narrow down the dataset to only Tregs.
    a. Create a Treg-specific subset.
    b. Re-process the Treg Subset.

```
so3 <- subset(so, subset = SingleR6 == "T cells (T.Tregs)")

# Re-process so3

so3 <- FindNeighbors(so3, reduction = "harmony", dims = 1:20)

so3 <- RunUMAP(so3, reduction = "harmony", dims = 1:20, reduction.name =
"umap.harmony")

so3 <- FindClusters(so3, resolution = 0.5, cluster.name = "harmony_clusters")

saveRDS(so3, file.path(path, "int.treg.rds"))
```

44. Generate key visualizations (focused on T cells/Tregs) (Figures 5C and 5D). More representative visualizations can be found in Xie et al.[1]
    a. Generate proportion Bar Plots (Cluster Composition Across Samples).
    b. Generate Gene Expression Visualizations (Feature Plot + Violin Plot).

```
# Proportion Bar Plots
```

```
create_prop_plot <- function(obj, filename) {

  v <- as.data.frame(table(obj$orig.ident, obj$harmony_clusters))

  p <- ggplot(v, aes(fill = Var2, y = Freq, x = Var1)) +

geom_bar(position = "fill", stat = "identity") +

    theme_bw() + labs(x = "Sample", y = "Proportion", fill = "Cluster") +

    theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggsave(filename, plot = p, width = 6, height = 6)

}

create_prop_plot(so2, file.path(path, "harmony2_clusterprop.pdf"))

create_prop_plot(so3, file.path(path, "harmony3_clusterprop.pdf"))

# Gene Feature Plots

genes_of_interest <- c("Nt5e", "Ccr6", "Cxcr4", "Foxp3", "Ikzf4", "Ikzf2")

f1 <- FeaturePlot(so2, reduction = "umap.harmony", features = genes_
of_interest,

        order = TRUE, min.cutoff = "q1", max.cutoff = "q99", split.by =
"orig.ident")

ggsave(file.path(path, "perSample_features.1.pdf"), plot = f1, width = 18,
height = 24)

v1 <- VlnPlot(so2, features = genes_of_interest, ncol = 2,

       group.by = "harmony_clusters", pt.size = 0)

ggsave(file.path(path, "perSample_violin.1.pdf"), plot = v1, width = 12,
height = 12)
```

45. Differential Expression (DE) analysis to identify genes that are regulated by Eos in thymic Tregs.
    a. Global DE Analysis in all Tregs (Eos cKO vs. WT).
    b. DE analysis for each individual Treg cluster (0-17) to find genotype-specific changes unique to that cluster.

```
so2 <- readRDS(file.path(path, "int.filtered2.rds")) # Integrated Full Object

so3 <- readRDS(file.path(path, "int.filtered3.rds")) # Treg Subset

# Overall cKO vs WT in Tregs

Idents(so3) <- "mutant"
```

```
de_all <- FindMarkers(so3, ident.1 = "Ikzf4 cKO", ident.2 = "WT", logfc.
threshold = 0.5, min.pct = 0.05, test.use = "MAST")

de_all$gene <- rownames(de_all)

write.csv(de_all, file.path(path, "cKOvsWT_allcells.csv"), quote = FALSE,
row.names = FALSE)

# By-cluster DE Analysis (Looping through clusters 0-17)

de_by_cluster <- list()

for (i in 0:17) {

  cluster_sub <- subset(so2, subset = harmony_clusters.1 == i)

  if(length(unique(cluster_sub$mutant)) == 2) { # Ensure both groups exist in
cluster

    Idents(cluster_sub) <- "mutant"

    temp_de <- FindMarkers(cluster_sub, ident.1 = "Ikzf4 cKO", ident.2 = "WT",

                logfc.threshold = 0.5, min.pct = 0.05, test.use = "MAST")

    if(nrow(temp_de) > 0) {

      temp_de$gene <- rownames(temp_de)

      temp_de$cluster <- i

      de_by_cluster[[paste0("c", i)]] <- temp_de

    }

  }

}

de_all_clusters <- bind_rows(de_by_cluster)

write.csv(de_all_clusters, file.path(path, "cKOvsWT_bycluster.csv"), quote =
FALSE, row.names = FALSE)
```

46. TCR repertoire analysis to characterize immune repertoire diversity, V-gene usage, and sample similarity (Figures 5E and 5F, 5G and 5H).
    a. Generate a V-gene Frequency plot showing the frequency of different V-genes across groups defined.
    b. Calculate and plot immune repertoire diversity for each group.
    c. Create a circular (Circos) plot to visualize overlap of clonotypes between groups.

```
# V-gene Frequency
```

```
p1 <- plot_clone_frequency(so2, data_col = "v_gene", cluster_col = "orig.
ident", panel_scales = "free")

ggsave(file.path(path, "vgene_split.pdf"), p1, width = 10, height = 6)

# Diversity Indices (Shannon/Simpson)

p2 <- plot_diversity(so2, data_col = "clonotype_id", cluster_col =
"orig.ident",

          method = list(shannon = abdiv::shannon, simpson = abdiv::simpson))

ggsave(file.path(path, "repetoire_split.pdf"), p2, width = 10, height = 6)

# Circos Overlap & MDS

p3 <- plot_similarity(so2, data_col = "clonotype_id", cluster_col =
"orig.ident",

          group_col = "mutant", method = "circos", scale = TRUE)

p4 <- plot_mds(so2, data_col = "clonotype_id", cluster_col = "orig.ident")

ggsave(file.path(path, "overlap.pdf"), p3, width = 6, height = 6)

ggsave(file.path(path, "mds.pdf"), p4, width = 6, height = 6)
```

47. Pathway Enrichment Visualizations. Representative visualizations are provided in Xie et al.[1]
    a. Generate a combined bubble/bar plot for pathway enrichment results.
    b. Load data and generate final plot.

```
#Helper function to process enrichment data and generate bubble/bar plots

plot_enrichment <- function(df, title) {

  df %>%

    mutate(

    `Enrichment FDR` = as.numeric(as.character(`Enrichment FDR`)),

    `Fold Enrichment` = as.numeric(as.character(`Fold Enrichment`)),

    log2FE = log2(`Fold Enrichment`),

    log2FE = if_else(Direction == "Downregulated", -log2FE, log2FE)

    ) %>%

    ggplot(aes(x = log2FE, y = Pathway, fill = -log10(`Enrichment FDR`), col =
-log10(`Enrichment FDR`))) +

    geom_point(aes(size = nGenes)) +
```

```
    geom_bar(stat = "identity", width = 0.1) +
    scale_fill_gradient(low = "blue", high = "red") +
    scale_color_gradient(low = "blue", high = "red") +
    theme_bw() +
    theme(text = element_text(size = 20)) +
    labs(x = "Log2 Fold Enrichment", y = "Pathway", title = title)
}
# Load and Plot Enrichment
enrich_all <- read_excel(file.path(path, "DE_cKOvsWT.xlsx"), sheet =
"cKOvsWT_allcells_enrichment")
p_enrich <- plot_enrichment(enrich_all, "Ikzf4 cKO vs WT - Global") +
facet_grid(Database ~ ., scales = "free_y", space = "free_y")
ggsave(file.path(path, "cKOvsWT_enrichment.pdf"), p_enrich, width = 20,
height = 15)
```

48. Define a reusable R function called plot_volcano that automatically creates a volcano plot.
    a. Define reusable R function.
    b. Take DE gene results as input.
    c. Classify genes into 3 groups.
    d. Label significant genes directly on the plot.
    e. Draw a clean volcano plot.
    f. Use geom_text_repel so gene labels don't overlap.

```
# Reusable Volcano Logic
plot_volcano <- function(de_data, p_thresh = 1.3e-06, fc_thresh = 0.5) {
  de_data <- de_data %>%
    mutate(
      Significant = case_when(
      avg_log2FC > fc_thresh & p_val < p_thresh ~ "cKO Upregulated",
      avg_log2FC < -fc_thresh & p_val < p_thresh ~ "cKO Downregulated",
      TRUE ~ "Not Significant"
      ),
      label = if_else(Significant != "Not Significant", gene, NA_character_)
```

```
    )

  ggplot(de_data, aes(x = avg_log2FC, y = -log10(p_val), color = Significant)) +

    geom_point(alpha = 0.5) +

    geom_text_repel(aes(label = label), max.overlaps = 20) +

    scale_color_manual(values = c("cKO Downregulated" = "#377EB8", "cKO
Upregulated" = "#E41A1C", "Not Significant" = "grey")) +

    theme_bw() +

    geom_vline(xintercept = c(-fc_thresh, fc_thresh), linetype = "dashed") +

    geom_hline(yintercept = -log10(p_thresh), linetype = "dashed")

}
```

49. Visualize the expression of important Treg genes. Representative figures are provided in Xie et al.[1]
    a. Create Treg marker gene heatmap.
    b. Create Violin Plot for key immune genes.
    c. Create Feature Plot for co-stimulatory molecules.

```
# Treg Gene Heatmap (Faceted by CellGroup)

treg_data = as.data.frame(AverageExpression(so2, group.by =c("mutant","cellgroup"),
features =c("Penk","Areg","Tff1"))$RNA)

treg_data$Gene = row.names(treg_data)

treg_data = melt(as.data.frame(treg_data),id.vars="Gene")

treg_data$Sample   =   str_split_fixed(treg_data$variable,pattern="_",n=2)
[,1]

treg_data$CellGroup  =  str_split_fixed(treg_data$variable,pattern="_",n=2)
[,2]

colnames(treg_data)[2] ="Expression"

treg_data$Expression2 = log10(treg_data$Expression * 2000 + 1),

p_heat <- ggplot(treg_data %>% filter(Sample %in% c("F1-WT", "F2-WT")),

         aes(x = Gene, y = CellGroup, fill = Expression2)) +

  geom_tile() +

  scale_fill_gradientn(colours = inferno(50)[-c(1:5)]) +

  theme_minimal() + labs(title = "WT Treg Markers by Group")
```

```
# Multi-Feature Plots

key_features <- c("Il2ra","Nr4a1","Nr4a3","Ccr8","Irf4","Tnfrsf18")

v1 <- VlnPlot(so2, features = key_features, ncol = 3, pt.size = 0, log = TRUE)

f1 <- FeaturePlot(so3, features = c("Icos","Tigit"), split.by = "mutant", or-
der = TRUE)

ggsave(file.path(path, "precursor_vlnLog.pdf"), v1, width = 15, height = 6)

ggsave(file.path(path, "Icos_Tigit_feature.pdf"), f1, width = 8, height = 8)
```

50. Subset high-quality Treg clusters and labels four functional groups.
    a. Load the data.
    b. Generate a trajectory-specific subset.
    c. Classify cells into four Treg groups using cluster-based marker annotation.

```
so2 <- readRDS(file.path(path, "int.filtered2.rds"))

# Create so3 (Subsetting for Trajectory)

so3 <- subset(so2, subset = harmony_clusters.1 %in% 0:12)

saveRDS(so2, file.path(path, "int.filtered2.rds"))

saveRDS(so3, file.path(path, "int.filtered3.rds"))

# Define Cell Groups based on marker clusters

# (0,7,8,9,14) -> CD73-

# (1,3,4) -> CD73+ Ccr6- Cxcr4-

# (13,15) -> CD73+ Ccr6+ Cxcr4-

# (2,5,6,12,16,17) -> CD73+ Ccr6+ Cxcr4+

so2$cellgroup <- case_when(

  so2$harmony_clusters.1 %in% c(0,7,8,9,14) ~ "CD73-",

  so2$harmony_clusters.1 %in% c(1,3,4) ~ "CD73+ Ccr6- Cxcr4-",

  so2$harmony_clusters.1 %in% c(13,15) ~ "CD73+ Ccr6+ Cxcr4-",

  so2$harmony_clusters.1 %in% c(2,5,6,12,16,17) ~ "CD73+ Ccr6+ Cxcr4+",

  TRUE ~ "Foxp3-"

)
```

51. Execute a standard single-cell developmental trajectory workflow.
    a. Use Slingshot to infer developmental lineages and pseudotime trajectories.[24]
    b. Use tradeSeq to model gene expression changes along those trajectories.[25]
    c. Identify statistically significant dynamic genes that drive cellular differentiation/development along the inferred paths.

```
dimred <- so3@reductions$umap.harmony@cell.embeddings

clustering <- as.factor(so3$harmony_clusters.1)

# Calculate Lineages

lineages <- getLineages(data = dimred, clusterLabels = clustering)

curves <- getCurves(lineages, approx_points = 300, thresh = 0.01,

          stretch = 0.8, allow.breaks = FALSE, shrink = 0.99)

# Expression Modeling (tradeSeq)

counts <- so3@assays$RNA@layers$counts

filt_counts <- counts[rowSums(counts > 5) > (ncol(counts)/100), ]

sce <- fitGAM(counts = as.matrix(filt_counts), sds = curves)

# Association Test

pseudotime_association <- associationTest(sce) %>%

  mutate(fdr = p.adjust(pvalue, method = "fdr"),

      feature_id = rownames(.)) %>%

  arrange(pvalue).
```

52. Perform pairwise differential gene expression analysis.
    a. Set cell identities.
    b. Define a MAST differential expression function.
    c. Run pairwise comparisons between Treg subsets, saving results as CSV files.

```
# Differential Expression (MAST)

Idents(so2) <- "cellgroup"

# Comparison helper function

run_mast <- function(obj, ident1, ident2, filename) {

de <- FindMarkers(obj, ident.1 = ident1, ident.2 = ident2,

l         ogfc.threshold = 0.5, min.pct = 0.05, test.use = "MAST")
```

```
write.csv(de, filename, quote = FALSE)

}

run_mast(so3, "CD73-", "CD73+/CCR6-/CXCR4-", file.path(path, "de_cd73neg
Vcd73pos.csv"))

run_mast(so2, "CD73-", "CD73+/CCR6+/CXCR4-", file.path(path, "de_cd73neg
Vcd73posccd6pos.csv"))

run_mast(so3, "CD73-", "CD73+/CCR6+/CXCR4+", file.path(path, "de_cd73neg
Vcd73posccd6poscxcr4pos.csv"))

run_mast(so3, "CD73-", "Foxp3-", file.path(path, "de_cd73negVunknown.csv"))
```

53. Define a plotting function and generate pathway enrichment visualizations for Treg subset comparisons.
    a. Define a reusable R function to create and save pathway enrichment plots from enrichment results.
    b. Generate and export customized visualizations for all Treg subset comparisons.

```
# Reusable Function for Pathway Plots

plot_pathway_enrichment <- function(path, file_name, title, out_name) {

df <- read.csv (file.path(path, file_name)) %>%

    mutate(

    `Enrichment FDR` = as.numeric(`Enrichment FDR`),

    `Fold Enrichment` = as.numeric(`Fold Enrichment`),

    log2FE = log2(`Fold Enrichment`),

    log2FE = if_else(Direction == "Downregulated", -1 * log2FE, log2FE)

    )

  p <- ggplot(df, aes(x = log2FE, y = Pathway, fill = -log10(`Enrichment FDR`))) +

    geom_bar(stat = "identity", width = 0.1, aes(color = -log10(`Enrichment
 FDR`))) +

    geom_point(aes(size = nGenes, color = -log10(`Enrichment FDR`))) +

    scale_fill_gradient(low = "blue", high = "red") +

    scale_color_gradient(low = "blue", high = "red") +

    theme_bw() +

    facet_grid(Database ~ ., space = "free", scales = "free") +
```

```
    labs(title = title, x = "Log2 Fold Enrichment", y = "Gene Pathway", size =
"Gene Count") +

    theme(text = element_text(size = 20))

  ggsave(p, filename = file.path(path, out_name), width = 20, height = 20, dpi =
"retina")

}

# Generate all pathway plots

plot_pathway_enrichment(path, "de_cd73negVunknown.csv", "CD73- vs Foxp3-",
"cd73NvU_e.png")

plot_pathway_enrichment(path,"de_cd73negVcd73pos.csv", "CD73- vs CD73+",
"cd73Nvcd73P_e.png")

plot_pathway_enrichment(path,"de_cd73negVcd73posccd6pos.csv", "CD73- vs
CD73+/CCR6+", "cd73Nvcd73PC6P_e.png")

plot_pathway_enrichment(path,"de_cd73negVcd73posccd6poscxcr4pos.csv",
"CD73- vs Triple Pos", "cd73Nvcd73PC6PC4P_e.png")
```

54. Create QC violin plots for marker genes and prepare data for Slingshot lineage trajectory analysis.
    a. Generate violin plots for quality control (QC) of key Treg marker gene expression across clusters.
    b. Prepare and format data for Slingshot lineage and trajectory analysis of Treg cell developmental paths.

```
# Final QC Visuals

v1 <- VlnPlot(so2, features = c("Nt5e","Ccr6","Cxcr4","Foxp3"), pt.size = 0,

       group.by = "harmony_clusters.1", ncol = 4) +

theme(text = element_text(size = 18))

ggsave(file.path(path, "perSample_violin3A.pdf"), plot = v1, width = 20,
height = 5)

#1. Prepare Slingshot Input

dimred <- so3@reductions$umap.harmony@cell.embeddings

clustering <- as.factor(so3$harmony_clusters.1)

groups <- as.factor(so3$cellgroup)

# Run Slingshot Lineages

lineages_clustering <- getLineages(data = dimred, clusterLabels = clustering)
```

```
# Optional: lineages_clustering9 <- getLineages(., start.clus = '9')
```

55. Create a function that paints custom black lines and directional arrows onto the UMAP.
    a. Draw thick black Slingshot lineage curves.
    b. Create a tiny helper function m() to find the center point of any cluster on the UMAP.
    c. Draw 11 fixed directional arrows that define a specific, pre-determined developmental hierarchy between clusters.
    d. Make the arrows reusable across multiple plots.

```
# 2. Define Custom Drawing Function

# This avoids repeating the 11 'Arrows' calls for different plots

draw_manual_trajectory <- function(lineage_data) {

  lines(SlingshotDataSet(lineage_data), lwd = 3, col = 'black')

  # Helper to get mean of coordinates for specific clusters

  m <- function(clus_ids) colMeans(dimred[clustering %in% clus_ids, 1:2])

  # Restoration of your specific directional arrows

  Arrows(x0=m(11)[1], y0=m(11)[2], x1=mean(c(m(11)[1], m(10)[1])), y1=mean
(c(m(11)[2], m(10)[2])), arr.lwd=2)

  Arrows(x0=m(10)[1], y0=m(10)[2], x1=mean(c(m(0)[1], m(10)[1])), y1=mean
(c(m(0)[2], m(10)[2])), arr.lwd=2)

  Arrows(x0=m(0)[1], y0=m(0)[2], x1=mean(c(m(0)[1], m(9)[1])), y1=mean
(c(m(0)[2], m(9)[2])), arr.lwd=2)

  Arrows(x0=m(9)[1], y0=m(9)[2], x1=mean(c(m(1)[1], m(9)[1])), y1=mean
(c(m(1)[2], m(9)[2])), arr.lwd=2)

  Arrows(x0=m(7)[1], y0=m(7)[2], x1=mean(c(m(7)[1], m(8)[1])), y1=mean
(c(m(7)[2], m(8)[2])), arr.lwd=2)

  Arrows(x0=m(1)[1], y0=m(1)[2], x1=m(c(1,2))[1], y1=m(c(1,2))[2], arr.lwd=2)

  Arrows(x0=m(2)[1], y0=m(2)[2], x1=mean(c(m(2)[1], m(5)[1])), y1=mean
(c(m(2)[2], m(5)[2])), arr.lwd=2)

  Arrows(x0=m(2)[1], y0=m(2)[2], x1=mean(c(m(2)[1], m(12)[1])), y1=mean
(c(m(2)[2], m(12)[2])), arr.lwd=2)

  Arrows(x0=m(12)[1], y0=m(12)[2], x1=mean(c(m(6)[1], m(12)[1])), y1=mean
(c(m(6)[2], m(12)[2])), arr.lwd=2)

  Arrows(x0=m(1)[1], y0=m(1)[2], x1=mean(c(m(4)[1], m(1)[1])), y1=mean
(c(m(4)[2], m(1)[2])), arr.lwd=2)
```

```
  Arrows(x0=m(4)[1], y0=m(4)[2], x1=m(c(4,3))[1], y1=m(c(4,3))[2], arr.lwd=2)

}
```

56. Generate Slingshot Plots, which can be seen in Xie et al.[1] Troubleshooting 6.
    a. Define custom color palettes for cell clusters and experimental groups.
    b. Generate a 2-panel PDF plot focused on cell clusters overlaid on dimensionality reduction (dimred) space (UMAP/t-SNE).
    c. Generate a plot nearly identical to Plot 1, but colors cells by experimental groups and adjusts dot size for visibility.

```
pal_clusters <- c(RColorBrewer::brewer.pal(9, "Set1"), RColorBrewer::brewer.
pal(8, "Set2"))

pal_groups <- c("#F8766D", "#A3A500", "#00B0F6", "#E76BF3")

# Plot 1: Cluster-based Trajectory

pdf(file.path(path, "slingshot_lineage.pdf"), width = 12, height = 6)

par(mfrow=c(1,2))

# Left Panel: Points + Cluster Labels

plot(dimred[,1:2], col = pal_clusters[clustering], cex=.5, pch = 16,
main="Clusters")

for(i in levels(clustering)){

text(mean(dimred[clustering==i,1]), mean(dimred[clustering==i,2]), labels =
i,font = 2, cex=2)

}

# Right Panel: Points + Manual Arrows

plot(dimred[,1:2], col = pal_clusters[clustering], cex=.5, pch = 16)

draw_manual_trajectory(lineages_clustering)

dev.off()

# Plot 2: Group-based Trajectory (2A)

pdf(file.path(path, "slingshot_lineage.groups2A.pdf"), width = 12, height = 6)

par(mfrow=c(1,2))

# Left Panel: Group Labels

plot(dimred[,1:2], col = pal_groups[groups], cex=1.5, pch = 16, cex.axis=1.5,
main="Groups")
```

```
for(i in levels(groups)){

text(mean(dimred[groups==i,1]), mean(dimred[groups==i,2]), labels = i,
font = 2, cex = 1.5)

}

# Right Panel: Groups + Manual Arrows

plot(dimred[,1:2], col = pal_groups[groups], cex=1.5, pch = 16, cex.axis=1.5)

draw_manual_trajectory(lineages_clustering)

dev.off()
```

57. Generate VDJ analysis plots.
    a. Define paths and samples.
    b. Load VDJ data via loop.
    c. Combine TCR contigs into a single list object.

```
# Define paths and samples

samples <- c("F1_Ikzf4_cKO_thymic_Tregs", "F1_WT_thymic_Tregs",

       "F2_Ikzf4_cKO_thymic_Tregs", "F2_WT_thymic_Tregs")

# Load VDJ data via loop

tcr_data <- list()

for (s in samples) {

  csv_path <- file.path(path, s, "outs/per_sample_outs", s, "vdj_t/
filtered_contig_annotations.csv")

  tcr_data[[s]] <- read.csv(csv_path)

}

#Combine TCR contigs into a single list object

combined.TCR <- combineTCR(tcr_data, samples = samples)

saveRDS(combined.TCR, file.path(path, "combinedTCR.rds"))
```

58. Generate Clonal Quantification and Abundance plots (Figures 6A and 6B).
    a. Generate clonal quantification plots for different calling methods.
    b. Calculate clonal abundance from a combined TCR dataset, generates two versions of the plot (unscaled vs. scaled).

```
# Generate clonal quantification plots for different calling methods

q_calls <- list(strict = "Gene + NT", gene = "Gene", nt = "NT", aa = "Amino Acid")
```

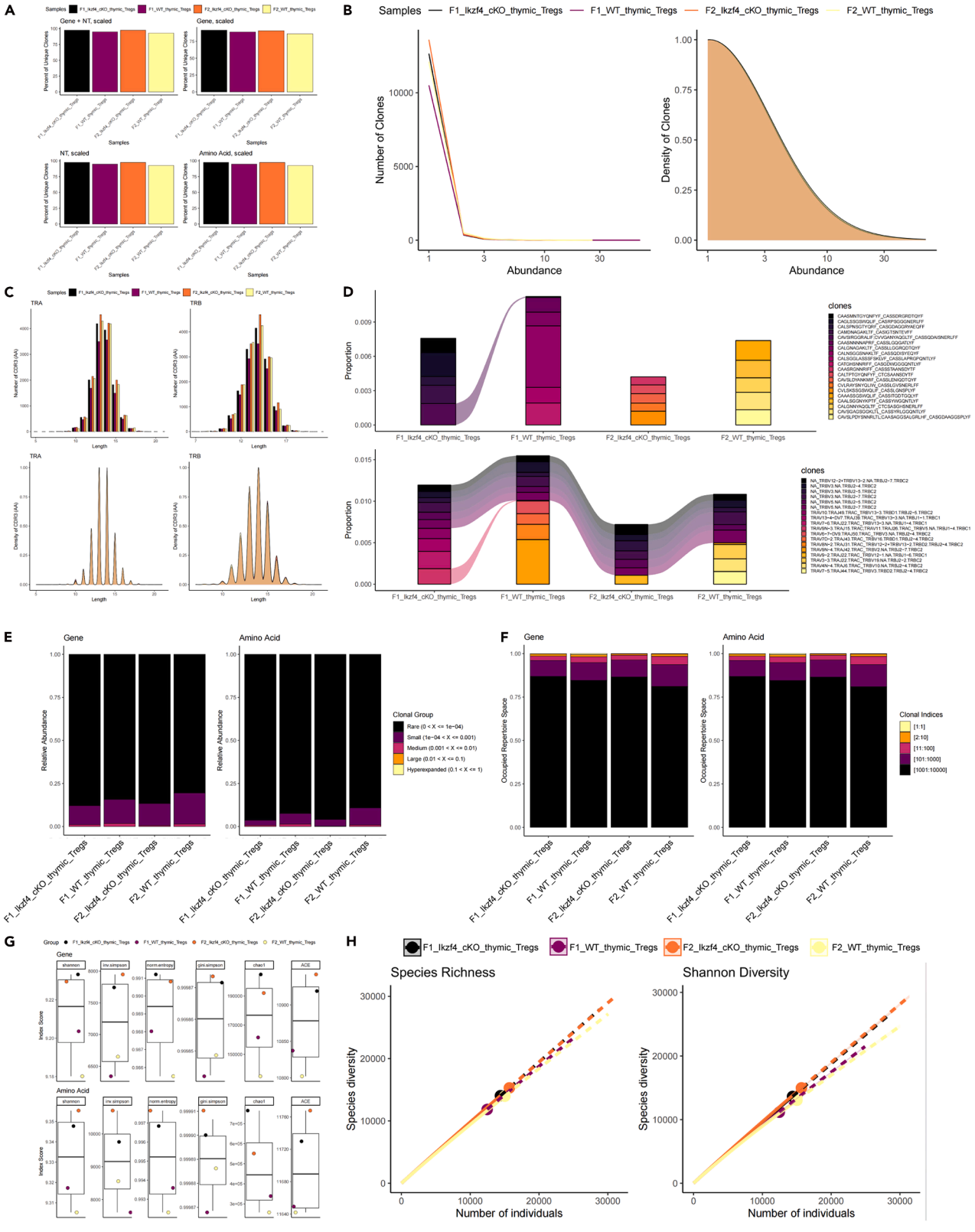


Figure 6. TCR repertoire analysis reveals clonal dynamics and diversity during TCR-seq computational analysis

(A) Clonal quantification plots comparing four clonotype calling methods (Gene + NT, Gene, NT, Amino Acid), with scaled read counts.
(B) Unscaled (left) and scaled (right) clonal abundance profiles using Gene-based clonotype calling.
(C)TCR CDR3 sequence length distributions. Histograms showing the length of CDR3 regions for TCRα and TCRβ chains in Ikzf4 cKO and WT thymic Treg samples, reflecting the diversity of the T cell receptor repertoire.
(D)Alluvial plots tracking the top 5 most abundant T cell clones across samples, comparing amino acid–based and gene-based clonotype calling.
(E)Clonal homeostasis plots illustrating the proportional space occupied by clones of distinct sizes, using gene- and amino acid–based clonotype calling.
(F)Clonal proportion distributions across sample groups, compared using nucleotide (NT) and amino acid (AA) clonotype definitions.
(G)TCR repertoire diversity analysis using gene and amino acid clonotype calling across experimental samples.
(H) Rarefaction curves estimating species richness and Shannon diversity of the TCR clonal repertoire.

```
q_plots <- lapply(names(q_calls), function(call) {

clonalQuant(combined.TCR, cloneCall = call, chain = "both", scale = TRUE) +

  labs(title = paste(q_calls[[call]], ", scaled")) +

  theme(axis.text.x = element_text(angle = 45, hjust = 1))

})

q_final <- ggarrange(plotlist = q_plots, common.legend = TRUE)

ggsave(file.path(path, "clonal_quant.pdf"), plot = q_final, width = 8,
height = 8)

# Clonal Abundance (Unscaled vs Scaled)

a1 <- clonalAbundance(combined.TCR, cloneCall = "gene", scale = FALSE)

a2 <- clonalAbundance(combined.TCR, cloneCall = "gene", scale = TRUE)

ggsave(file.path(path, "clonal_abundance.pdf"), plot = ggarrange(a1, a2,
common.legend = TRUE), width = 8, height = 4)
```

59. Analyze clonal length (of TRA/TRB chains) and clonal tracking across samples (Figures 6C and 6D).
    a. Compare length distributions for TRA (alpha chain) and TRB (beta chain) TCRs, with both unscaled and scaled versions.
    b. Visualize how TCR clones are distributed (tracked) across different samples.

```
# Compare TRA vs TRB lengths

l1 <- clonalLength(combined.TCR, cloneCall = "aa", chain = "TRA") + labs(ti-
tle = "TRA")

l2 <- clonalLength(combined.TCR, cloneCall = "aa", chain = "TRA", scale = TRUE) +
labs(title = "TRA Scaled")

l3 <- clonalLength(combined.TCR, cloneCall = "aa", chain = "TRB") + labs(ti-
tle = "TRB")
```

```
l4 <- clonalLength(combined.TCR, cloneCall = "aa", chain = "TRB", scale = TRUE) +
labs(title = "TRB Scaled")

ggsave(file.path(path, "clonal_length.pdf"), plot = ggarrange(l1, l3, l2, l4,
ncol = 2, nrow = 2, common.legend = TRUE), width = 10, height = 10)

# Alluvial plots to compare clonal tracking across samples

c1 <- clonalCompare(combined.TCR, top.clones = 5, samples = samples, clone-
Call = "aa", graph = "alluvial")

c2 <- clonalCompare(combined.TCR, top.clones = 5, samples = samples, clone-
Call = "gene", graph = "alluvial")

ggsave(file.path(path, "clonal_compare.pdf"), plot = ggarrange(c1, c2, ncol =
1), width = 12, height = 6)
```

60. Analyze clonal homeostasis (clone size distribution), clonal proportion (clone frequency), and clonal diversity/rarefaction (repertoire richness) (Figures 6E and 6F, 6G and 6H).
    a. Analyze clonal homeostasis (space occupied by clones of different sizes).
    b. Analyze clonal proportion.
    c. Analyze diversity and rarefaction (estimating richness).

```
# Clonal Homeostasis (Space occupied by clones of different sizes)

h1 <- clonalHomeostasis(combined.TCR, cloneCall = "gene") + labs(title =
"Gene")

h2 <- clonalHomeostasis(combined.TCR, cloneCall = "aa") + labs(title =
"Amino Acid")

ggsave(file.path(path, "clonal_Homeostasis.pdf"), plot = ggarrange(h1, h2,
common.legend = TRUE), width = 10, height = 5)

# Clonal Proportion

p1 <- clonalProportion(combined.TCR, cloneCall = "nt") + labs(title = "NT")

p2 <- clonalProportion(combined.TCR, cloneCall = "aa") + labs(title = "AA")

ggsave(file.path(path, "clonal_Proportion.pdf"), plot = ggarrange(p1, p2,
common.legend = TRUE), width = 10, height = 5)

# Diversity & Rarefaction (Estimating richness)

d1 <- clonalDiversity(combined.TCR, cloneCall = "gene")

d2 <- clonalDiversity(combined.TCR, cloneCall = "aa")

ggsave(file.path(path, "clonalDiversity.pdf"), plot = ggarrange(d1, d2, ncol =
1), width = 8, height = 8)
```

```
r1 <- clonalRarefaction(combined.TCR, hill.numbers = 0, n.boots = 2) + labs
(title = "Species Richness")

r2 <- clonalRarefaction(combined.TCR, hill.numbers = 1, n.boots = 2) + labs
(title = "Shannon Diversity")

ggsave(file.path(path, "clonalRichness.pdf"), plot = ggarrange(r1, r2,
common.legend = TRUE), width = 8, height = 4)
```

61. Visualize and save feature and violin plots for key marker genes across cell clusters and groups.
    a. Define marker gene panels.
    b. Create a reusable function to generate UMAP feature plots.
    c. Produce violin plots to visualize and save expression patterns of marker genes across cell clusters and groups.

```
#Define sets of markers for cleaner plotting

chemokine_markers <- c("Nt5e","Ccr3","Ccr4","Ccr7","Ccr8","Cxcr5","Ccr2",
"Ccr6","Ccr9","Cxcr3","Cxcr4","Cxcr6")

key_treg_markers <- c("Ikzf4","Foxp3","Ccr6")

itreg_ntreg_markers <- c("Ikzf2","Entpd1","Dpp4","Il2ra","Cd83","March1")

# Feature Plots for so2 (Total) and so3 (Tregs)

# Using a function to reduce repetition

save_feature_plot <- function(obj, feats, filename, w = 12, h = 16, nc = 3) {

fp <- FeaturePlot(obj, features = feats, order = TRUE, min.cutoff = "q1",

        ncol = nc, label = TRUE, reduction = "umap.harmony")

ggsave(filename, plot = fp, width = w, height = h, dpi = "retina")

}

save_feature_plot(so2, chemokine_markers, file.path(path, "marker_feature.pdf"))

save_feature_plot(so3, chemokine_markers, file.path(path, "treg_marker_
feature.pdf"))

save_feature_plot(so2, key_treg_markers, file.path(path, "key_marker_
feature.pdf", h = 4))

save_feature_plot(so3, key_treg_markers, file.path(path, "treg_key_marker_
feature.pdf", h = 4))

# iTreg vs nTreg Markers (Helios/Ikzf2 etc)

save_feature_plot(so2, itreg_ntreg_markers, file.path(path, "itreg_ntreg_
markers.pdf"), h = 8)
```

```
# Violin plots for itreg/ntreg markers by cluster and cell group

v1 <- VlnPlot(so2, features = itreg_ntreg_markers, ncol = 2, group.by =
"harmony_clusters.1", pt.size = 0)

v2 <- VlnPlot(so2, features = itreg_ntreg_markers, ncol = 2, group.by =
"cellgroup", pt.size = 0)

ggsave(file.path(path, "ntreg_violin_clusters.pdf"), plot = v1, width = 12,
height = 12)

ggsave(file.path(path, "ntreg_violin_groups.pdf"), plot = v2, width = 8,
height = 12)
```

Optional: To validate key markers identified by scRNA-seq analysis at the protein level, perform flow cytometric analysis of isolated thymic Tregs.

EXPECTED OUTCOMES

In this protocol thymic Tregs are FACS-sorted from 6 wo $Eos^{fl/fl}$ x Foxp3-YFPCre/RFP mice (2 replicates: F1/F2) for downstream scRNA-seq and TCR-seq. 60,000 to 100,000 WT or Eos cKO thymic Tregs can be isolated from each mouse (Figure 1). 4 10x GEM-X 5′ V3 gene expression libraries and 4 10x Genomics Single Cell VDJ libraries were sequenced on two NovaSeq X Plus runs. For the gene expression libraries, all libraries have sequencing yields of more than 869 million reads per library. The sequencing run was setup as a 28-cycle + 90-cycle non-symmetric run. Demultiplexing was done allowing one mismatch in the barcodes. The analysis was performed with the Cell Ranger 8.0.1 software using the default parameters. For each sample the number of cells captured ranges from 14,798 to 18,359 and mean reads per cell ranges from 47,381 to 61,848 (Figure 3). Cells with extremely low number of UMI counts were filtered out. Median genes found per cell ranges from 3,144 to 3,249 and the total number of genes detected ranges from 22,212 to 22,539 (Figure 3). For the 4 10x Genomics Single Cell VDJ libraries, all libraries have sequencing yields of more than 145 million reads per library. The number of cells captured ranges from 12,466 to 15,685 and mean reads per cell ranges from 10,392 to 12,210 (Figure 3). Cells with productive V-J spanning pair ranges from 11,291 to 14,188 (Figure 3). Paired clonotype diversity is between 7,599.08 and 13,805.02 (Figure 3).

The scRNA-seq and TCR VDJ analysis pipeline for studying recirculating thymic Tregs and the regulatory function of Eos is split into four core steps (Figure 4). Step 1 covers initial QC and data integration: it loads 10x Genomics gene expression (GEX) and VDJ data into a Seurat object, calculates mitochondrial QC metrics, filters low-quality cells (nFeature_RNA 200–6000, percent.mt <10%), and integrates data via Harmony and CCA to correct batch effects. SingleR (ImmGen reference) annotates cell types, followed by subsetting to T cells (so2) and pure Tregs (so3), with UMAP clustering and visualizations (violin/scatter/feature plots) for key markers (Figure 5). 17 clusters are identified in the T cell subset (so2).[1] Step 2 identifies genes differentially expressed using MAST, comparing Eos cKO vs WT in all Tregs and individual T cell clusters (logFC >0.5, min.pct=0.05), and outputs DE gene list. It also performs TCR repertoire analysis (V-gene frequency, Shannon/Simpson diversity, circos/MDS similarity) and pathway enrichment, with reusable functions for volcano and enrichment bubble plots, plus violin/feature plots for Treg activation markers (Figure 5).[1] This step identified genes and pathways regulated by Eos within Tregs and individual T cell clusters, shedding light on the regulatory function of Eos. Step 3 conducts Slingshot trajectory analysis, stratifying Tregs into four cell groups by CD73/Ccr6/Cxcr4 expression ($CD73^-$ to $CD73^+Ccr6^+Cxcr4^+$). It infers maturation lineages, models gene expression over pseudotime with tradeSeq, runs DE on trajectory subsets, and generates custom trajectory plots with manual directional arrows, plus high-resolution pathway enrichment plots for cell group

comparisons. These findings indicate that CD73$^+$ cells exhibit a more mature phenotype relative to CD73$^-$ cells and highlight the presence of distinct developmental programs within the CD73$^+$Ccr6$^-$Cxcr4$^-$ and CD73$^+$Ccr6$^+$Cxcr4$^+$ populations. Step 4 deepens VDJ analysis: it loads and combines TCR contigs, quantifies clonality via multiple calling methods (Gene, NT, AA), and visualizes clonal abundance, CDR3 length (TRA/TRB), and sharing (alluvial plots) (Figure 6). It also assesses clonal homeostasis, proportion, diversity, and rarefaction, with additional Seurat feature/violin plots for chemokine receptors and iTreg/nTreg markers (Figure 6). This step confirms that CD73$^+$Ccr6$^+$Cxcr4$^+$ cells have a highly distinct clonal repertoire relative to the other four groups, implying that these cells are recirculating thymic Tregs whose TCR repertoire has been re-shaped by peripheral antigen exposure. Combined with step 3, the results identify that CD73$^+$Ccr6$^+$Cxcr4$^+$ thymic Tregs are predominantly recirculating thymic Tregs.

Together, this integrated experimental and computational framework enables systematic characterization of thymic Treg maturation, clonal dynamics, and the transcriptional regulatory function of Eos.

LIMITATIONS

This protocol has several key limitations. FACS sorting of thymic Tregs is vulnerable to variability in tissue dissociation, cell viability, and instrument performance, which may introduce transcriptional artifacts or reduce capture efficiency. Only two biological replicates per genotype limit statistical power and reproducibility. Environmental factors—including temperature fluctuations during tissue handling, inconsistent library preparation, and subtle NovaSeq X Plus sequencing variations—may introduce unresolved batch effects. Fixed QC and differential expression thresholds could exclude rare cell populations or functionally relevant subtle transcriptional changes. Bioinformatically, reliance on ImmGen annotation, clustering parameters, and Slingshot pseudotime inference may misclassify subsets or skew maturation trajectory directionality. TCR-VDJ analysis is constrained by incomplete paired chain recovery, underestimating clonal overlap. Finally, the protocol is optimized for the analysis of Tregs from the thymus of steady-state young mice and may not reliably resolve recirculating Tregs in inflamed, aged, or other tissues.

TROUBLESHOOTING

Problem 1

Insufficient thymic Treg yield.

Potential solution

- Carefully inspect the thoracic cavity after excision to ensure the entire thymus (two lobes) is removed.
- Rinse the Petri dish and cell strainer with 2–3 mL of collection medium to recover residual cells.
- Use strictly 6 wo mice—avoid mice older than 7 weeks.
- Pool cells from 2–3 mice if a higher yield is needed for downstream steps.

Problem 2

Low GEM generation efficiency (<80%).

Potential solution

- Filter cell suspension through a 70 μm strainer immediately before loading into the GEM-X Chip.
- Follow 10x Genomics instructions strictly for chip loading—verify all reagents are added in the correct order.
- Update Chromium X/iX firmware to ≥2.0.0 before running.
- Recheck cell concentration and adjust to 1300–1600 cells/μL before loading.
- Use freshly prepared Master Mix and avoid bubbles when loading the chip.

Problem 3

Poor library QC (Bioanalyzer peaks are broad or noisy).

Potential solution

- Follow 10x Genomics protocol for cDNA amplification and cleanup—ensure all centrifugation steps are performed at the correct speed/time.
- Maintain strict sterile technique during library construction; use RNase-free reagents and equipment.
- Verify fragmentation parameters (time/temperature) according to the 5′ Reagent Kits v3 instructions.
- Use fresh adaptors and ensure ligation reaction is incubated at the correct temperature (as specified by the protocol).
- Dilute libraries 1:10 before Bioanalyzer QC to avoid signal saturation.

Problem 4

Low sequencing yield per library (<869 million reads for GEX libraries).

Potential solution

- Quantify libraries using a Qubit or similar instrument before pooling—ensure equal molarity for each library.
- Set sequencing run to 28-cycle + 90-cycle non-symmetric (as recommended for 10x GEM-X 5′ V3 libraries).
- Use two independent quantification methods (e.g., Qubit + Bioanalyzer) to verify library concentration.
- Follow Illumina instructions for NovaSeq X Plus flow cell loading—avoid bubbles and ensure proper sealing.

Problem 5

Unreliable cell type annotation with SingleR.

Potential solution

- Re-download the ImmGen reference dataset and verify it is loaded correctly in R.
- Join Seurat layers (as described in Step 41a) before running SingleR.
- Reapply filtering criteria (nFeature_RNA 200–6000, percent.mt <10%) to remove low-quality cells.
- Normalize and scale gene expression data in Seurat before running SingleR annotation.
- Use the "fine" annotation mode for more specific cell type labels, then clean up labels (remove extra text in parentheses).

Problem 6

Slingshot trajectory analysis fails to infer lineages.

Potential solution

- Re-process the Treg subset to increase cluster number (adjust resolution parameter in Seurat).
- Re-run UMAP with different dimensions (e.g., 20–30 PCs) to improve separation.
- Generate feature plots for CD73, Ccr6, and Cxcr4 to verify cell group stratification—adjust gates if needed.
- Update the slingshot package and its dependencies (e.g., tradeSeq) in R.
- Use custom color palettes for cell clusters to improve visualization of trajectory paths.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Ethan M. Shevach (eshevach@niaid.nih.gov).

Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contact, Xuan Xie (xxsabrina0408@outlook.com).

Materials availability

This study did not generate new reagents.

Data and code availability

- The datasets in this study are available on GEO: GSE305537.
- The code from this study is fully listed in this article and as supplemental material.

ACKNOWLEDGMENTS

This research was supported by the Intramural Research Program of the National Institute of Allergy and Infectious Disease, National Institutes of Health (NIH) (award no.: AI000959-20). The contributions of the NIH authors were made as part of their official duties as NIH federal employees, are in compliance with agency policy requirements, and are considered Works of the United States Government. However, the findings and conclusions presented in this paper are those of the authors and do not necessarily reflect the views of the NIH or the U.S. Department of Health and Human Services.

We thank the Genomic Research Section (RTB, NIAID) for performing scRNA-seq and TCR-seq as well as primary data processing. We thank the Integrated Data Sciences Section (RTB, NIAID) for the computational analysis. We thank the Flow Cytometry Section (RTB, NIAID, NIH) for cell sorting.

AUTHOR CONTRIBUTIONS

X.X. designed and conducted this study and wrote the manuscript. T.H. analyzed the sequencing data and wrote the analytical code. F.O.-C. performed the scRNA-seq and TCR-seq experiments. X.Z. revised the manuscript. E.M.S. secured funding for the study and revised the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work, the authors used artificial intelligence (AI) to check for grammatical errors. After using this tool or service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2026.104620.

REFERENCES

1. Xie, X., Thornton, A.M., Tanwar, S., Korty, P., Wells, A.C., Segrist, E., Teijeiro, A., Rivera, C.A., Wei, D., Chen, X., et al. (2026). Eos plays a critical role in Treg homeostasis and modulates the function of recirculating thymic Tregs in the control of Treg development. Cell Rep. *45*, 116838. https://doi.org/10.1016/j.celrep.2025.116838.

2. Josefowicz, S.Z., Lu, L.F., and Rudensky, A.Y. (2012). Regulatory T cells: mechanisms of differentiation and function. Annu. Rev. Immunol. *30*, 531–564. https://doi.org/10.1146/annurev.immunol.25.022106.141623.

3. Sakaguchi, S., Yamaguchi, T., Nomura, T., and Ono, M. (2008). Regulatory T cells and immune tolerance. Cell *133*, 775–787. https://doi.org/10.1016/j.cell.2008.05.009.

4. Thiault, N., Darrigues, J., Adoue, V., Gros, M., Binet, B., Perals, C., Leobon, B., Fazilleau, N., Joffre, O.P., Robey, E.A., et al. (2015). Peripheral regulatory T lymphocytes recirculating to the thymus suppress the development of their precursors. Nat. Immunol. *16*, 628–634. https://doi.org/10.1038/ni.3150.

5. Cowan, J.E., Baik, S., McCarthy, N.I., Parnell, S.M., White, A.J., Jenkinson, W.E., and Anderson, G. (2018). Aire controls the recirculation of murine Foxp3+ regulatory T-cells back to the thymus. Eur. J. Immunol. *48*, 844–854. https://doi.org/10.1002/eji.201747375.

6. Weist, B.M., Kurd, N., Boussier, J., Chan, S.W., and Robey, E.A. (2015). Thymic regulatory T cell niche size is dictated by limiting IL-2 from antigen-bearing dendritic cells and feedback competition. Nat. Immunol. *16*, 635–641. https://doi.org/10.1038/ni.3171.

7. Lio, C.-W.J., and Hsieh, C.-S. (2008). A Two-Step Process for Thymic Regulatory T cell

development. Immunity *28*, 100–111. https://doi.org/10.1016/j.immuni.2007.11.021.

8. Owen, D.L., Mahmud, S.A., Sjaastad, L.E., Williams, J.B., Spanier, J.A., Simeonov, D.R., Ruscher, R., Huang, W., Proekt, I., Miller, C.N., et al. (2019). Thymic regulatory T cells arise via two distinct developmental programs. Nat. Immunol. *20*, 195–205. https://doi.org/10.1038/s41590-018-0289-6.

9. Lemarquis, A.L., Kousa, A.I., Argyropoulos, K.V., Jahn, L., Gipson, B., Pierce, J., Serrano-Marin, L., Victor, K., Kanno, Y., Girotra, N.N., et al. (2025). Recirculating regulatory T cells mediate thymic regeneration through amphiregulin following damage. Immunity *58*, 397–411.e6. https://doi.org/10.1016/j.immuni.2025.01.006.

10. Pan, F., Yu, H., Dang, E.V., Barbi, J., Pan, X., Grosso, J.F., Jinasena, D., Sharma, S.M., McCadden, E.M., Getnet, D., et al. (2009). EOS mediates FOXP3-Dependent gene silencing in CD4 + regulatory T cells. Science *325*, 1142–1146. https://doi.org/10.1126/science.1176077.

11. Santamaria, J.C., Borelli, A., and Irla, M. (2021). Regulatory T cell heterogeneity in the thymus: impact on their functional activities. Front. Immunol. *12*, 643153. https://doi.org/10.3389/fimmu.2021.643153.

12. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of Single-Cell data. Cell *177*, 1888–1902.e21. https://doi.org/10.1016/j.cell.2019.05.031.

13. Wickham, H., François, R., Henry, L., Müller, K., and Vaughan, D. (2026). dplyr: A Grammar of Data Manipulation. R package version 1.2.1. https://dplyr.tidyverse.org.

14. Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis (Springer-Verlag). https://ggplot2.tidyverse.org.

15. Borcherding, N., Bormann, N.L., and Kraus, G. (2020). scRepertoire: An R-based toolkit for single-cell immune receptor analysis. F1000Res. *9*, 47. https://doi.org/10.12688/f1000research.22139.2.

16. Sheridan, R. (2023). djvdj: A collection of single-cell V(D)J tools. https://rnabioco.github.io/djvdj/. https://github.com/rnabioco/djvdj/.

17. Wickham, H. (2025). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.6.0. https://stringr.tidyverse.org.

18. Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-R., and Raychaudhuri, S. (2019). Fast, sensitive and accurate integration of single-cell data with Harmony. Nat. Methods *16*, 1289–1296. https://doi.org/10.1038/s41592-019-0619-0.

19. Aran, D., Looney, A.P., Liu, L., Wu, E., Fong, V., Hsu, A., Chak, S., Naikawadi, R.P., Wolters, P.J., Abate, A.R., et al. (2019). Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. Nat. Immunol. *20*, 163–172. https://doi.org/10.1038/s41590-018-0276-y.

20. Slowikowski, K. (2026). ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'. R package version 0.9.8. https://ggrepel.slowkow.com/.

21. Wickham, H., and Bryan, J. (2025). readxl: Read Excel Files. R package version 1.4.5. https://github.com/tidyverse/readxl. https://readxl.tidyverse.org.

22. Kassambara, A. (2026). ggpubr: 'ggplot2' Based Publication Ready Plots. R package version 0.6.3. https://rpkgs.datanovia.com/ggpubr/.

23. Garnier, S., Ross, N., Rudis, R., Camargo, P.A., Sciaini, M., and Scherer, C. (2024). viridis(Lite) - Colorblind-Friendly Color Maps for R. Zenodo. https://doi.org/10.5281/zenodo.4679423. https://sjmgarnier.github.io/viridis/.

24. Street, K., Risso, D., Fletcher, R.B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S. (2018). Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genom. *19*, 477. https://doi.org/10.1186/s12864-018-4772-0.

25. Van Den Berge, K., Roux de Bézieux, H., Street, K., Saelens, W., Cannoodt, R., Saeys, Y., Dudoit, S., and Clement, L. (2020). Trajectory-based differential expression analysis for single-cell sequencing data. Nat. Commun. *11*, 1201. https://doi.org/10.1038/s41467-020-14766-3.

26. Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat. Biotechnol. *36*, 411–420. https://doi.org/10.1038/nbt.4096.

27. Heng, T.S.P., Painter, M.W., Lukacs-Kornek, V., Mauermann, N., Turley, S.J., Koller, D., Kim, F.S., Wagers, A.J., and Asinovski, N. (2008). The Immunological Genome Project: networks of gene expression in immune cells. Nat. Immunol. *9*, 1091–1094. https://doi.org/10.1038/ni1008-1091.