



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA QUÍMICA, BIOTECNOLOGÍA Y
MATERIALES

**EXPLORING THE IMPACTS OF A NOVEL TARGETED
IMMUNOTHERAPY ON TUMOR-INFILTRATING IMMUNE CELLS BY
SINGLE CELL RNA SEQUENCING (SCRNA-SEQ).**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL EN BIOTECNOLOGÍA

SOFÍA CRISTINA CAVADA SILVA

PROFESORA GUÍA:

MARÍA ORIANA SALAZAR AGUIRRE

MIEMBROS DE LA COMISIÓN:

JOSHUA WATERFALL

ALVARO OLIVERA NAPPA

SANTIAGO DE CHILE

2024

RESUMEN DE LA MEMORIA PARA OPTAR AL TÍTULO
DE INGENIERA CIVIL EN BIOTECNOLOGÍA
POR: SOFÍA CRISTINA CAVADA SILVA
FECHA: 2024
PROFESORA GUÍA: MARÍA ORIANA SALAZAR AGUIRRE

EXPLORACIÓN DE LOS EFECTOS DE UNA NUEVA INMUNOTERAPIA DIRIGIDA EN CÉLULAS INMUNITARIAS INFILTRANTES DE TUMORES MEDIANTE SECUENCIACIÓN DE ARN UNICELULAR (SCRNA-SEQ).

Los tumores rabdoides (RTs) corresponden a tumores infantiles altamente agresivos, caracterizados principalmente por la inactivación de SMARCB1, componente central del complejo de remodelación de cromatina SWI/SNF, y por poseer una carga mutacional baja. Históricamente se ha supuesto que los RTs son invisibles para el sistema inmunitario y, en consecuencia, malos candidatos para la inmunoterapia. Sin embargo, en trabajos recientes realizados en el Institut Curie y en otros estudios, se ha demostrado que los RTs son inmunogénicos y que tanto los RTs humanos como los de ratón están altamente infiltrados por células mieloides y células T CD8+. Mediante bloqueo de la vía PD-1/PDL-1, también conocido como tratamiento anti-PD1, se generó una disminución en el crecimiento tumoral, aunque no se logró eliminar por completo los tumores. Por ello hoy en día se busca potenciar la respuesta antitumoral combinando el tratamiento anti-PD1 con otras terapias para eliminar estos tumores.

Mediante scRNA-Seq, se determinó un tratamiento que teóricamente podría estar dirigido a poblaciones mieloides, induciendo un retraso significativo del crecimiento tumoral en los ratones portadores de tumores tratados. Por ello El objetivo de este trabajo fue estudiar el efecto de dicho tratamiento sobre lasmieloides utilizando scRNA-seq para comprender mejor el mecanismo de acción subyacente. Para llevar a cabo este análisis, se concedió acceso a conjuntos de datos de scRNA-seq de ratones tratados y no tratados generados en el Institut Curie y se procesaron en R studio siguiendo pipelines bioinformáticos especializados.

Los resultados obtenidos mostraron que el tratamiento incrementa las poblaciones de granulocitos intratumorales a la vez que disminuye las poblaciones de células dendríticas, monocitos y macrófagos totales. En concreto, se observó un aumento de macrófagos que expresan Cxcl9, junto con una disminución de las subpoblaciones de células dendríticas convencionales (cDCs), indicando ambos un posible aumento del reclutamiento de células T, que fue confirmado mediante un estudio de poblaciones de células T CD8+. Al mismo tiempo, aumentaron las subpoblaciones de células dendríticas tipo 3 (DC3) y de células Supresoras Granulocíticas-Mieloides (G-MDSCs), ambas de fenotipo inmunosupresor, lo que agotaría a las células T reclutadas. Por ende, se recomienda aplicar este tratamiento junto con tratamientos que eviten el agotamiento de las células T, como anti-PD1, para obtener un mejor efecto.

Para mi mamá Irlanda, mi abuela Zoila y mi abuelo Juan, quienes me criaron y me apoyaron para poder lograr llegar hasta acá.

Acknowledgement

I would like to express my deepest gratitude to the University where I have been able to train and acquire the necessary tools for my future career. Though these years were not without challenges, especially with the pandemic, I received a solid education thanks to the dedication of my professors. It has allowed me to advance in my academic and professional path.

Special thanks to the Institut Curie, for giving me the opportunity to be part of their research. Thanks to Joshua Waterfall, for his invaluable supervision and guidance, and Valeria Manríquez for opening the doors to this opportunity and for her continuous support throughout the process. Her confidence in me has been an inspiring driving force.

I cannot fail to mention my friends, who have accompanied me throughout these university years, some even from the very first year. Thank you for brightening my days, for your unconditional love and for being a constant support on this journey.

Finally, many thanks to my family. To my mother and grandparents, who have lovingly raised and guided me and without whom I would never have even been able to enter university. To my uncles, aunts and cousins who have always believed in my abilities and have been a constant support. My only regret is that my grandfather was not able to witness the culmination of this stage, I hope I have fulfilled the faith he placed in me.

To each and every one of you, thank you for being a fundamental part of my journey and for contributing to my personal and professional growth.

Table of Content

| | |
|--|-----------|
| 1. Introduction | 1 |
| 1.1. Objectives | 2 |
| 1.1.1. Specific objectives | 3 |
| 1.2. Limitations and Scope | 3 |
| 2. State of the Art | 4 |
| 2.1. Current treatments in Rhabdoid tumors | 4 |
| 2.2. Tumor infiltration of immune cells in Rhabdoid tumors | 5 |
| 2.2.1. Monocytes and Macrophages | 5 |
| 2.2.2. Dendritic cells | 7 |
| 2.2.3. Granulocytes | 8 |
| 2.3. Bioinformatic analysis of scRNA-seq | 9 |
| 3. Methodology and Materials | 12 |
| 4. Results and Discussions | 14 |
| 4.1. Dendritic cells | 24 |
| 4.2. Monocyte and Macrophages | 29 |
| 4.3. Granulocytes | 36 |
| 5. Conclusions | 49 |
| 6. Future Developments | 51 |
| Bibliography | 52 |
| ANNEXES | 61 |
| A. Previous results | 61 |
| B. Mice datasets analysis code | 63 |
| C. Dendritic cells mice analysis code | 114 |
| D. Monocyte/Macrophages mice analysis code | 155 |
| E. Granulocyte cells mice analysis code | 199 |
| F. Log2 Fold Change | 241 |
| F.1. General myeloid data | 241 |

| | | |
|------|---|-----|
| F.2. | Dendritic cell data | 241 |
| F.3. | Monocyte and Macrophages data | 242 |
| F.4. | Granulocytes data | 242 |
| F.5. | Neutrophils data | 242 |
| G. | CD8+ T cell subsets analysis | 243 |

Index of Tables

| | | |
|------|--|-----|
| F.1. | Number of cells in the different myeloid populations under untreated and treated conditions, alongside their log ₂ FC value. | 241 |
| F.2. | Number of cells in the different myeloid populations under untreated and treated conditions, alongside their log ₂ FC value. | 241 |
| F.3. | Number of cells in the different dendritic cells subpopulations under untreated and treated conditions, alongside their log ₂ FC value. | 241 |
| F.4. | Number of cells in the different monocyte and macrophage subpopulations under untreated and treated conditions, alongside their log ₂ FC value. | 242 |
| F.5. | Number of cells in the different granulocyte subpopulations under untreated and treated conditions, alongside their log ₂ FC value. | 242 |
| F.6. | Number of cells in the different neutrophil subpopulations under untreated and treated conditions, alongside their log ₂ FC value. | 242 |

Index of Illustrations

| | | |
|-------|--|----|
| 3.1. | scRNA-seq analysis flowchart | 13 |
| 4.1. | General data pipeline | 15 |
| 4.2. | Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 16 |
| 4.3. | Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 17 |
| 4.4. | Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 17 |
| 4.5. | Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 18 |
| 4.6. | Merged data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples. | 19 |
| 4.7. | Integrated data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples. | 19 |
| 4.8. | Integrated data clustering. a) Integrated data UMAP resolution 0.1. b) Integrated data Clustree. | 20 |
| 4.9. | Cd3e analysis. a) Cd3e expression level. b) Cd3e violin plot. | 20 |
| 4.10. | Integrated data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples. | 21 |
| 4.11. | Myeloid cells classification. a) UMAP divided by populations. b) Key genes expression analysis: Mmp9, Csf1r and Flt3 | 21 |
| 4.12. | Myeloid key genes Dotplot | 22 |
| 4.13. | Myeloid populations changes analysis. a) Barplot of populations percentage divided by Condition. b) Myeloid populations UMAPs divided by Sample. . . | 23 |
| 4.14. | Dendritic cells pipeline | 24 |
| 4.15. | Dendritic cells data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples. | 25 |
| 4.16. | Dendritic cells clustering. a) Dendritic cells UMAP resolution 0.2. b) Dendritic cells Clustree. | 25 |

| | | |
|-------|--|----|
| 4.17. | Dendritic cells Heatmaps. a) cDC1 signatures. b) DC3 signature. c) pDC signatures. d) cDC2 signatures. e) MoDC signatures. | 26 |
| 4.18. | Dendritic cells subpopulations. | 27 |
| 4.19. | Dendritic cells subpopulations changes. a) Barplot of dendritic cells subpopulations percent by Conditions. b) Untreated dendritic cells density plots. c) Treated dendritic cells density plots | 27 |
| 4.20. | Monocyte/Macrophages pipeline | 29 |
| 4.21. | Monocyte/Macrophages data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples. | 30 |
| 4.22. | Monocyte/Macrophages clustering. a) Monocyte/Macrophages UMAP resolution 0.2. b) Monocyte/Macrophages Clustree. | 30 |
| 4.23. | Monocyte/Macrophages resolution 0.7 Dotplot | 31 |
| 4.24. | Key genes expression analysis | 31 |
| 4.25. | Monocyte/Macrophages trajectory using Monocle3 | 32 |
| 4.26. | Monocyte/Macrophages subpopulations | 33 |
| 4.27. | Monocyte/Macrophages subpopulations changes. a) Barplot of monocyte/macrophages subpopulations percent by Conditions. b) Untreated monocyte/macrophages density plots. c) Treated monocyte/macrophages density plots | 33 |
| 4.28. | Cxcl9 and Spp1 expression analysis in macrophages. a) Cxcl9 Expression in macrophages. b) Cxcl9 expression in Untreated macrophages. c) Cxcl9 expression in Treated macrophages. d) Cxcl9 violin plots divided by Condition. e) Spp1 Expression in macrophages. f) Spp1 expression in Untreated macrophages. g) Spp1 expression in Treated macrophages. h) Spp1 violin plots divided by Condition. | 34 |
| 4.29. | Cxcl10 expression analysis in macrophages. a) Cxcl10 Expression in macrophages. b) Cxcl10 expression in Untreated macrophages. c) Cxcl10 expression in Treated macrophages. d) Cxcl10 violin plots divided by Condition | 35 |
| 4.30. | Mreg subpopulation analysis. a) Comparison of iNOS expression in each macrophage cluster under treated and untreated conditions. b) Upregulated pathways in Mreg cluster under untreated conditions. c) Upregulated pathways in Mreg cluster after treatment condition in comparison with the untreated condition. | 35 |
| 4.31. | Granulocytes pipeline | 36 |
| 4.32. | Granulocytes data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples. | 37 |
| 4.33. | Granulocytes clustering. a) Granulocytes UMAP resolution 0.1. b) Granulocytes Clustree. | 37 |
| 4.34. | Zilionis mNeutrophils signature analysis. a) mNeutrophils signature expression level. b) mNeutrophils signature violin plot | 38 |
| 4.35. | Granulocytes resolution 0.1 Dotplot | 38 |

| | | |
|-------|---|----|
| 4.36. | Alshetaiwi Transcriptionally active neutrophils signature analysis. a) Transcriptionally active neutrophils signature expression level. b) Transcriptionally active neutrophils signature violin plot | 39 |
| 4.37. | Gurtner eosinophils key genes signature analysis. a) Eosinophils key genes expression level. b) Eosinophils key genes violin plot | 40 |
| 4.38. | Granulocytes subpopulations | 41 |
| 4.39. | Granulocytes subpopulations changes. a) Barplot of granulocytes subpopulations percent by Conditions. b) Untreated granulocytes density plots. c) Treated granulocytes density plots | 41 |
| 4.40. | Neutrophils data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples. | 42 |
| 4.41. | Neutrophils clustering. a) Neutrophils UMAP resolution 0.1. b) Neutrophils Clustree. | 42 |
| 4.42. | Neutrophil trajectory using Monocle 2. a) Neutrophils trajectory colored by State. b) Neutrophils trajectory colored by Pseudotime. c) Neutrophils trajectory colored by Seurat Clusters. d) Neutrophils trajectory divided by Seurat Clusters. | 44 |
| 4.43. | Alshetaiwi mouse G-MDSCs signature analysis. a) Mouse G-MDSCs expression level. b) Mouse G-MDSCs expression level on Monocle2 trajectory graphic . . | 45 |
| 4.44. | Monocle2 data on UMAP. a) Neutrophils Monocle2 States. b) Neutrophils Monocle2 Pseudotime | 46 |
| 4.45. | Alshetaiwi Progenitor neutrophil signature analysis. a) Progenitor neutrophil expression level. b) Progenitor neutrophil expression level on Monocle2 trajectory graphic | 46 |
| 4.46. | Neutrophils subpopulations | 47 |
| 4.47. | Neutrophils subpopulations changes. a) Barplot of neutrophils subpopulations percent by Conditions. b) Untreated neutrophils density plots. c) Treated neutrophils density plots | 48 |
| A.1. | Analysis of immune populations in the tumor microenvironment. Information provided by Institute Curie | 61 |
| A.2. | Analysis of Macrophage depletion in tumor growth. Information provided by Institute Curie | 61 |
| A.3. | Analysis of possible druggable targets and study of treatment effect in tumor growth. Information provided by Institute Curie | 62 |
| B.1. | Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 67 |
| B.2. | Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 67 |

| | | |
|-------|--|-----|
| B.3. | Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 68 |
| B.4. | Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 68 |
| B.5. | Elbow plot Merged data | 70 |
| B.6. | UMAP of Merged data divided by Condition | 71 |
| B.7. | UMAP of Merged data divided by Sample | 72 |
| B.8. | Elbow plot Integrated data | 74 |
| B.9. | UMAP of Integrated data divided by Sample | 75 |
| B.10. | UMAP of Integrated data divided by Condition | 76 |
| B.11. | Integrated data Clustree | 77 |
| B.12. | Integrated data UMAP resolution 0.1 | 78 |
| B.13. | Cd3e expression level | 79 |
| B.14. | Cd3e violin plot | 80 |
| B.15. | Elbow plot Integrated data | 83 |
| B.16. | UMAP of Integrated data divided by Sample | 84 |
| B.17. | UMAP of Integrated data divided by Condition | 85 |
| B.18. | Integrated data Clustree | 86 |
| B.19. | Ptprc expression level | 87 |
| B.20. | Integrated data UMAP resolution 6.9 | 88 |
| B.21. | Ptprc violin plot | 88 |
| B.22. | Elbow plot Integrated data | 91 |
| B.23. | UMAP of Integrated data divided by Sample | 92 |
| B.24. | UMAP of Integrated data divided by Condition | 93 |
| B.25. | Mmp9 expression level | 94 |
| B.26. | Csf1r expression level | 95 |
| B.27. | Flt3 expression level | 96 |
| B.28. | Myeloid key genes Dotplot | 97 |
| B.29. | Granulocytes manual selection | 98 |
| B.30. | Granulocytes | 99 |
| B.31. | Granulocytes divided by Condition | 100 |
| B.32. | Dendritic cells manual selection | 101 |
| B.33. | Dendritic cells | 102 |
| B.34. | Dendritic cells divided by Condition | 103 |
| B.35. | Monocyte/Macrophages manual selection part 1 | 104 |
| B.36. | Monocyte/Macrophages manual selection part 2 | 105 |
| B.37. | Monocyte/Macrophages | 106 |
| B.38. | Monocyte/Macrophages divided by Condition | 107 |
| B.39. | Classified populations UMAP | 108 |

| | | |
|-------|--|-----|
| B.40. | Density plots | 111 |
| B.41. | Barplots of myeloid cells populations number by Condition | 112 |
| B.42. | UMAPs of myeloid cells populations divided by Sample | 113 |
| C.1. | Histogram of the number of features expressed vs number of cells for Dendritic cells Untreated sample 1 | 115 |
| C.2. | Histogram of the number of features expressed vs number of cells for Dendritic cells Untreated sample 2 | 115 |
| C.3. | Histogram of the number of features expressed vs number of cells for Dendritic cells Treated sample 1 | 116 |
| C.4. | Histogram of the number of features expressed vs number of cells for Dendritic cells Treated sample 2 | 116 |
| C.5. | Scatter plot of Dendritic cells Untreated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 119 |
| C.6. | Scatter plot of Dendritic cells Untreated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 119 |
| C.7. | Scatter plot of Dendritic cells Treated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 120 |
| C.8. | Scatter plot of Dendritic cells Treated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 120 |
| C.9. | Elbow plot Dendritic cells | 122 |
| C.10. | UMAP of Dendritic cells divided by Condition | 123 |
| C.11. | UMAP of Dendritic cells divided by Sample | 124 |
| C.12. | UMAP of Dendritic cells level of feature expression | 125 |
| C.13. | Manual selection of low features cells | 126 |
| C.14. | Elbow plot Dendritic cells | 128 |
| C.15. | UMAP of Dendritic cells divided by Condition | 129 |
| C.16. | UMAP of Dendritic cells divided by Sample | 130 |
| C.17. | Dendritic cells Clustree | 133 |
| C.18. | Dendritic cells density plots | 136 |
| C.19. | Integrated Dendritic cells UMAP resolution 0.2 | 137 |
| C.20. | Heatmap cDC1 Signatures | 140 |
| C.21. | Heatmap cDC2 Signatures | 144 |
| C.22. | Heatmap DC3 Signatures | 147 |
| C.23. | Heatmap pDC Signatures | 150 |
| C.24. | Heatmap MoDC Signatures | 152 |
| C.25. | Dendritic cells subpopulations | 153 |
| C.26. | Barplot of Dendritic cells subpopulations percent by Conditions | 154 |

| | | |
|-------|---|-----|
| D.1. | Histogram of the number of features expressed vs number of cells for Mono- cyte/Macrophages Untreated sample 1 | 156 |
| D.2. | Histogram of the number of features expressed vs number of cells for Mono- cyte/Macrophages Untreated sample 2 | 156 |
| D.3. | Histogram of the number of features expressed vs number of cells for Mono- cyte/Macrophages Treated sample 1 | 157 |
| D.4. | Histogram of the number of features expressed vs number of cells for Mono- cyte/Macrophages Treated sample 2 | 157 |
| D.5. | Scatter plot of Monocyte/Macrophages Untreated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 160 |
| D.6. | Scatter plot of Monocyte/Macrophages Untreated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 160 |
| D.7. | Scatter plot of Monocyte/Macrophages Treated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 161 |
| D.8. | Scatter plot of Monocyte/Macrophages Treated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values. | 161 |
| D.9. | Elbow plot Monocyte/Macrophages | 163 |
| D.10. | UMAP of Monocyte/Macrophages divided by Condition | 164 |
| D.11. | UMAP of Monocyte/Macrophages divided by Sample | 165 |
| D.12. | Monocyte/Macrophages Clustree | 168 |
| D.13. | Integrated Monocyte/Macrophages cells UMAP resolution 0.7 | 170 |
| D.14. | Mmp9 violin plot | 171 |
| D.15. | Contaminant granulocytes clusters | 172 |
| D.16. | New integrated Monocyte/Macrophages cells UMAP resolution 0.7 without con- taminants | 173 |
| D.17. | New Monocyte/Macrophages Clustree without contaminants | 174 |
| D.18. | Monocyte/Macrophages density plots | 177 |
| D.19. | Monocyte/Macrophages resolution 0.7 Dotplot | 178 |
| D.20. | Caption | 179 |
| D.21. | Monocyte/Macrophages trajectory using Monocle3 | 181 |
| D.22. | Monocyte/Macrophages subpopulations | 182 |
| D.23. | Barplot of Monocyte/Macrophages subpopulations percentage by Conditions . | 183 |
| D.24. | Macrophages subpopulations | 184 |
| D.25. | Spp1 expression level in Macrophages | 185 |
| D.26. | Spp1 expression level in untreated Macrophages | 186 |
| D.27. | Spp1 expression level in treated Macrophages | 187 |
| D.28. | Macrophages Spp1 violin plot by Condition | 188 |

| | | |
|-------|---|-----|
| D.29. | Cxcl9 expression level in Macrophages | 190 |
| D.30. | Cxcl9 expression level in untreated Macrophages | 191 |
| D.31. | Cxcl9 expression level in treated Macrophages | 192 |
| D.32. | Macrophages Cxcl9 violin plot by Condition | 193 |
| D.33. | Cxcl10 expression level in Macrophages | 195 |
| D.34. | Cxcl10 expression level in untreated Macrophages | 196 |
| D.35. | Cxcl10 expression level in treated Macrophages | 197 |
| D.36. | Macrophages Cxcl10 violin plot by Condition | 198 |
| E.1. | Elbow plot Granulocytes | 200 |
| E.2. | UMAP of Granulocytes divided by Condition | 201 |
| E.3. | UMAP of Granulocytes divided by Sample | 202 |
| E.4. | Granulocytes Clustree | 205 |
| E.5. | Granulocytes density plots | 208 |
| E.6. | Integrated Granulocytes UMAP resolution 0.1 | 209 |
| E.7. | Zilionis mNeutrophils signature expression level | 210 |
| E.8. | Zilionis mNeutrophils signature violin plot | 211 |
| E.9. | Granulocytes resolution 0.1 Dotplot | 212 |
| E.10. | Alshetaiwi Translationally active Neutrophils signature expression level | 213 |
| E.11. | Alshetaiwi Translationally active Neutrophils signature violin plot | 214 |
| E.12. | Gurtner Eosinophils key genes signature expression level | 215 |
| E.13. | Gurtner Eosinophils key genes signature violin plot | 216 |
| E.14. | Granulocytes subpopulations | 217 |
| E.15. | Barplot of Granulocytes subpopulations percentage by Conditions | 218 |
| E.16. | Elbow plot Neutrophils | 220 |
| E.17. | UMAP of Neutrophils divided by Condition | 221 |
| E.18. | UMAP of Neutrophils divided by Sample | 222 |
| E.19. | Neutrophils Clustree | 225 |
| E.20. | Neutrophils density plots | 228 |
| E.21. | Integrated Neutrophils UMAP resolution 0.7 | 229 |
| E.22. | Monocle2 Neutrophils trajectory colored by State | 231 |
| E.23. | Monocle2 Neutrophils trajectory colored by Pseudotime | 231 |
| E.24. | Monocle2 Neutrophils trajectory colored by Seurat Clusters | 232 |
| E.25. | Monocle2 Neutrophils trajectory divided by Seurat Clusters | 232 |
| E.26. | Neutrophils UMAP divided by Monocle2 States | 233 |
| E.27. | Pseudotime in Neutrophils UMAP | 234 |
| E.28. | Alshetaiwi’s Mouse G-MDSC signature expression on UMAP | 235 |
| E.29. | Alshetaiwi’s Mouse G-MDSC signature expression on Monocle trajectory | 236 |
| E.30. | Alshetaiwi’s Progenitor Neutrophil signature expression on UMAP | 237 |
| E.31. | Alshetaiwi’s Progenitor Neutrophil signature expression on Monocle trajectory | 238 |
| E.32. | Neutrophils subpopulations | 239 |
| E.33. | Barplot of Neutrophils subpopulations percentage by Conditions | 240 |

| | | |
|------|--|-----|
| G.1. | Analysis of T cell population changes under untreated and treated conditions. Information provided by Institute Curie | 243 |
| G.2. | CD8+ subpopulations change analysis under untreated and treated conditions | 244 |
| G.3. | Progenitor exhausted (Tpex) and terminally exhausted (Tex) CD8+ T cells change analysis under untreated and treated conditions.Information provided by Institute Curie | 244 |

1

Introduction

Rhabdoid Tumors (RTs) are a low-incidence but very aggressive type of pediatric tumor that can develop both in the brain (called Atypical teratoid/rhabdoid tumor or AT/RT) and in soft tissues throughout the body (Extra-cranial Malignant Rhabdoid Tumor or ECRT) [1]. It has been observed that they account for 1-2% of all pediatric brain tumors detected and, in the case of infants under 3 years of age, this percentage may increase to 10-20%. There is currently no established standard treatment, however, given the aggressiveness of these tumors, they require intense treatments such as surgery, chemotherapy and radiotherapy [1; 2], which are effective in a minority of patients and are frequently associated with severe sequelae, a factor of utmost relevance given the age of most patients [3]. Therefore, there is an urgent need to develop innovative treatments that are more efficient and generate fewer adverse effects.

RTs present two main characteristics. First, they are characterized by being genomically and genetically simple and presenting a biallelic inactivation of the gene coding for SMARCB1 (95% of cases) [4], which is a key protein of the SWI/SNF complex, responsible for remodelling chromatin through an ATP-dependent system [5; 6]. In other words, in the absence of SMARCB1, there are alterations in the chromatin state and deregulation of gene expression that lead the formation of tumors. In the remaining 5% of cases, these tumors could be formed by the inactivation of the SMARCA4 protein, another protein that is part of the SWI/SNF [4] complex.

Secondly, RTs are characterized by being one of the tumors with the lowest mutational load compared to all existing tumors [5]. This is relevant since the current paradigm of cancer in adult tumors suggests that those with high mutational load, by presenting a high expression of neoantigens by the tumor cells, can be recognized by the cells of the immune system and, together with this, make them more susceptible to immunotherapy treatments [7]. In the case of RTs, having a low mutational load, they have historically been considered invisible to the immune system and a poor candidate for immunotherapy.

However, recently, it has been found that, despite their low mutational burden, these tumors can be immunogenic and can respond to treatment with immunotherapy [5; 8]. Specifically, it has been observed that the inactivation of SMARCB1 in tumor cells generates a deregulation of gene expression, due to SWI/SNF deficiencies, promoting the expression of transposable elements such as endogenous retroviruses (ERVs). ERVs are normally silenced in healthy cells but can be activated in cancer cells due to epigenetic changes, leading to the generation of double-stranded RNA (dsRNA) within tumor cells, which can trigger an IFN signalling pathway, simulating a viral infection and thus favoring the infiltration of immune cells [3].

Immunotherapy based on the “immune checkpoints blockade” (ICB) such as CTLA-4 and PD-1/PDL1 has demonstrated its efficacy in the treatment of multiple cancers in adults. The host laboratory at Institut Curie demonstrated, using a well-established RT mouse model [9], that blockade of the PD1/PDL1 pathway also favoured the control of growth of this particular tumor, revealing RTs are indeed immunogenic and may be susceptible to immunotherapy. However, this treatment was not strong enough to eliminate the tumor in all cases [5]. For this reason, the current goal is to enhance the antitumor response by combining anti-PD1/anti-PDL1 with other therapies to achieve the elimination of RTs. Due to their high tumor-infiltration and pro-tumoral characteristics, myeloid cells present themselves as a promising target for therapeutic interventions.

Myeloid cells comprise a diverse set of cell populations that play crucial roles in the tumor microenvironment (TME). These are divided into mononuclear cells, which include macrophages and dendritic cells (DCs), and polymorphonuclear cells, corresponding to granulocytes such as neutrophils and eosinophils. Their role is multifaceted and highly complex, as they can have both “anti-tumor” effects by participating in the immune response, and “pro-tumor” effects by generating an immunosuppressive environment [10]. The role that these cells may play depends on the TME. Pro-tumoral effects encompass various mechanisms that contribute to tumor cell growth and invasion, the promotion of tumor angiogenesis and T-cell exhaustion [11]. On the other hand, they can also perform the exact opposite function, participating in the direct elimination of tumor cells, the production of angiostatic factors that inhibit the formation of new blood vessels, and the priming of T cells, bolstering the immune response against cancer [10; 11]. The plasticity that myeloid cells have in response to TME, change them in a good candidate for immunotherapy. In this work, we will evaluate the direct effect of a new treatment on myeloid cells and how this change can affect the recruitment and priming of cytotoxic T cells.

1.1. Objectives

Analyze the effects of novel targeted immunotherapy on tumor-infiltrating immune populations using scRNA-seq.

1.1.1. Specific objectives

- Determine the method for analysis of scRNA-seq data.
- Identify the different immune cell subpopulations present in the analysis, according to the literature.
- Determine changes in cell subpopulations by comparing qualitative and quantitative changes in the treated and untreated conditions.
- To analyze the correlation between changes in the size of tumor-infiltrating immune cell subpopulations and the effects of therapy, and to determine their significance in terms of response to treatment.

1.2. Limitations and Scope

The main focus of the study is exploring the impact of a novel immunotherapy in mouse tumor-infiltrating immune cells, using scRNAseq datasets generated in Institut Curie. Due to the characteristics of each immune compartment, it will be necessary to evaluate different bioinformatic pipelines for a correct analysis of the data. For example, neutrophils are cells that express a low numbers of genes, also known as features, and can be easily confused with low-quality cells, so in the myeloid analysis, it will be necessary to exclude neutrophils before removing low features cells. Also, to select a resolution and define the clusters, a combination between supervised (comparing our cluster with genetic signatures previously identified in the literature) and unsupervised analysis (analyzing the genetic signature of each cluster to identified them) will be used. The analysis will be based on the pipeline used at the Institute Curie and changes will be discussed with different institutional bioinformaticians for proper data management.

2

State of the Art

2.1. Current treatments in Rhabdoid tumors

As previously mentioned, rhabdoid tumours are a rare but highly aggressive form of pediatric cancer [1; 2]. These tumours are characterized by biallelic inactivation of SMARCB1, leading to deficiency of the SWI/SNF chromatin remodelling complex and tumour formation in the nervous system and soft tissues [2; 4; 5]. Despite their low incidence, RTs represent a major clinical challenge due to their highly aggressive behaviour and resistance to conventional therapies [1].

Currently, the treatment of rhabdoid tumours is based on multimodality therapeutic approaches, including surgery, chemotherapy and radiotherapy, however, the long-term survival rate for RT patients remains low [1]. This is because existing therapies for rhabdoid tumours have several limitations. On the one hand, the aggressiveness of these tumours makes them difficult to treat by surgery due to local invasion and tendency to metastasize [1]. In addition, RTs tend to be resistant to chemotherapy, a common feature in SWI/SNF-driven cancers, limiting the efficacy of conventional treatment regimens [3]. Moreover, radiotherapy, despite being potentially effective, can have significant side effects, especially in pediatric patients [1]. These limitations highlight the urgent need to develop more effective and better-targeted therapeutic approaches for rhabdoid tumours.

In this context, recent studies of the host laboratory have shown that rhabdoid tumours are infiltrated by both T cells and myeloid cells, indicating the presence of an active immune response. These findings open new opportunities for the development of immune therapies; however, it is important to note that the focus of immunotherapeutic studies has been primarily on T cells and their therapeutic potential in the treatment of rhabdoid tumours [5], without delving into the effects of immunotherapies on myeloid cells.

2.2. Tumor infiltration of immune cells in Rhabdoid tumors

Recent studies of the host laboratory have shown that rhabdoid tumours are infiltrated by both T cells and myeloid cells, indicating the presence of an active immune response. In the initial study, the focus was to study the characteristics of the tumor-infiltrating lymphocytes (TILs) and their therapeutic potential in the treatment of rhabdoid tumours [5], without delving into the effects of those different immunotherapies on myeloid cells.

A multiparametric approach involving FACS, immunohistochemistry and scRNAseq allowed the identification of at least three large groups of human myeloid cells on AT/RTs: Granulocytes, Monocytes/macrophages (MoMa) and dendritic cells (DCs). Currently the role of myeloid cells in tumours is an active and complex area of research, as these cells may have both cancer-promoting and cancer-suppressive functions, and their role may vary depending on the tumour type and the local microenvironment [12]. For example, in lung adenocarcinoma, breast cancer and colorectal carcinoma, it has been observed that myeloid cell abundance is associated with advanced cancer stage, decreased survival and worse clinical outcomes [11], while in nasopharyngeal cancer or lung cancer the presence of mast cells or granulocyte respectively were associated with a good prognosis. In this work, we will study in depth the characteristics of the myeloid infiltrate to understand how a treatment can induce changes in the different subpopulations, improving the control of tumor growth.

2.2.1. Monocytes and Macrophages

Monocytes, as implied by their name, are mononuclear cells comprising a heterogeneous population circulating in the bloodstream. They can be categorized into two distinct groups with different functions: resident monocytes (also known as non-classical monocytes) and classical monocytes. Non-classical monocytes are characterized by a low expression of Ly6c in mice [13] and are CD14⁻ CD16⁺ in humans [14]. This subpopulation resides in the blood vessel lumen, henceforth called resident monocytes in mice, and has been described as a patrolling population that senses and scans the endothelial surface for damage [13]. After the recognition of damage signals, they produce inflammatory cytokines, participate in the intravascular recruitment of neutrophils and can be involved in the clearance of debris after an inflammatory process [15]. On the other hand, classical monocytes are characterized by Ly6c high expression in mice [13] and CD14⁺ CD16⁻ in humans [14]. These cells are recruited to various organs and can act as precursors of peripheral mononuclear phagocytes such as macrophages and dendritic cells, which in response to tissue environmental cues will contribute to local inflammation or its resolution [13]. In steady state, some monocytes can extravasate to tissues without totally differentiating in mononuclear phagocytes. These

cells are known as tissue monocytes or monocyte-derived cells and are characterized by the expression of Ly6C+ MHC-II+ [16].

Previously it was believed that the homeostasis of tissue-resident macrophages relied mostly on the constant recruitment of blood Monocytes, the reason why these two populations are studied together. However, according to recent studies monocytes do not contribute to the maintenance of most peripheral tissue macrophage populations in adults in the steady state, these macrophages are seeded before birth and can maintain themselves in adults by self-renewal, independently of monocytes. According to these findings, monocyte-derived macrophages and monocyte-derived dendritic cells, which were also believed to be generated from monocytes in some cases, are now considered particular subpopulations between their peers [13].

Tumor associated macrophages (TAMs) are a heterogenous cell population of the TME that can account for up to 50% of solid tumors such as RTs [17]. The use of scRNAseq has revealed the high plasticity of this population, which can be classified according to their origin (monocyte-derived or yolk sac-derived), phenotypes (pro-tumoral, anti-tumoral) and the function they developed [18].

At least two different ways have been used to classify macrophages according to their phenotype in the context of cancer: M1/M2 classification or according to CXCL9/SPP1 expression. M1/M2 is a classic and simple classification to describe two major and opposing activities of macrophages. M1 activity, associated to an anti-tumoral activity, inhibits cell proliferation, causes tissue damage and is characterized by iNOS expression, while M2 activity, associated to a pro-tumoral activity, promote cell proliferation, tissue repair and is characterized by Arginase expression and IL-10 production [19; 20]. However, it has been observed that macrophage phenotypes are more complex and have greater plasticity than this simple definition would suggest [21]. The most updated studies have shown a new classification according to the expression of Cxcl9, a chemokine that participate in T-cells recruitment, or the expression of SPP1, an enzyme involved in tumor cell growth, migration and chemo-resistance. Therefore, evidence indicates that macrophages expressing Cxcl9 exhibit an anti-tumor phenotype, correlating with improved prognosis across various cancer types. In contrast, SPP1 macrophages demonstrate a pro-tumor phenotype and are linked to a poor prognosis [22].

According to their function, there are at least two types of macrophages present in all tumors, which show totally opposite phenotypes: IFN-macrophages and myeloid regulatory cells (mReg). IFN-macrophages are cells that, in response to interferon, promotes the production of inflammatory cytokines that contribute to an enhanced anti-tumor response [23]. On the other hand, mRegs are a recently discovered pro-tumoral population with charac-

teristics in between monocytes and macrophages, that participate in the promotion of tumor growth and the suppression of immune responses [24].

An exhaustive study of monocytes and macrophages in freshly collected samples from patients with rhabdoid tumors unveiled the prominence of the MoMa subpopulation, surpassing other immune cell types. Moreover, this study identified the presence of all previously described populations (pro and anti-tumoral) [5]. Nevertheless, the alterations of these populations in response to immunotherapy remains unexplored.

2.2.2. Dendritic cells

Just as monocytes and macrophages, dendritic cells correspond to mononuclear cells. They are highly specialized antigen-presenting cells that are found in various tissues throughout the body, including the skin, mucosal surfaces, and lymphoid organs [11]. Their current classification is mostly based on function, tissue localization and developmental origin [11; 12; 25].

Plasmacytoid dendritic cells, also known as pDC, correspond to a CD123+ subtype [25], characterized by their high interferon- α production in response to viral infection, activating cells from both the innate and adaptive immune response, such as natural killer cells and T cells [11]. On the other hand, dendritic cells that specialize in antigen sampling in tissues and migration to lymph nodes to induce antigen-specific T cell immunity are known as conventional dendritic cells [11]. These cells are further divided by the type of T cell they activate, with type 1 corresponding to CD8+ and type 2 to CD4+. Conventional dendritic cells type 1 (cDC1) are characterized by the expression of IRF8 and have a specialized function in the induction of CD8+ T cell responses. This enables them to initiate cytotoxic immune responses against intracellular pathogens and cancer cells. In preclinical models, cDC1 have also been shown to play a crucial role in the response to checkpoint blockade therapies and has been associated with improved overall survival in cancer patients [26; 27]. Regarding type 2 (cDC2), these are characterized by IRF4 expression and are mainly involved in CD4+ T cell activation, facilitating the generation of T helper cell responses that regulate various aspects of the immune system [26; 27].

Aside from these three dendritic cells subtypes other populations of notice are the previously mentioned monocyte-derived dendritic cells (MoDC), which appear as derivate from monocytes during inflammation [28; 29]. In cancer studies, MoDCs have proven to be difficult to differentiate from cDC2 cells, as these present similar surface markers and phenotypes, therefore, in various cases, these cell populations end up being grouped despite their distinct ontogeny [29]. Additionally, new DC subpopulations have been identified in both mice and human samples, including a LAMP3+ subpopulation [5], CCR7+ subpopulations [30],

immunoregulatory dendritic cells (mregDC) [31; 32] and an “activated” subpopulation, that has been observed in several studies, referred to as DC3 [21; 29; 33]

2.2.3. Granulocytes

Besides mononuclear immune cells, one can also find polymorphonuclear phagocytes, which are referred to as granulocytes. Granulocytes are characterized by the presence of granules in their cytoplasm, which contain various enzymes and proteins involved in immune defense. They can be further classified into multiple populations or subpopulations such as neutrophils, eosinophils, and basophils [11].

In the context of cancer studies, neutrophils are of particular interest due to their potential for both pro-tumoral and anti-tumoral phenotype, similar to macrophages [11; 21; 34]. In the past, tumour associated neutrophils (TANs) have been classified as N1/anti-tumorigenic and N2/protumorigenic. However, with the advancement of technologies like mass cytometry and scRNA-seq, it has been discovered that there is a high plasticity and heterogeneity within neutrophils in the context of cancer. As a result, recent studies suggest that a better way to classify neutrophils would be based on their developmental stage [34]. Based on their developmental stage, neutrophils can be classified as the following:

- Common myeloid progenitors (CMP): CMPs are multipotent hematopoietic stem cells that give rise to all myeloid lineages, including neutrophils, monocytes, and erythrocytes.
- Granulocyte-monocyte progenitors (GMP): GMPs are a subset of CMPs that are committed to the granulocyte and monocyte lineages. GMPs give rise to neutrophils, eosinophils, and basophils.
- Pro-neutrophils stage 1 (proNeu1) and stage 2 (proNeu2): Pro-neutrophils are a developmental stage of neutrophils that are intermediate progeny and do not expand during emergency granulopoiesis.
- Neutrophil precursors (preNeu): Pre-neutrophils are highly proliferative, poorly motile precursors that drive expansion within the spleen and bone marrow.
- Immature neutrophils (immNeu): Immature neutrophils are neutrophils that have been released from the bone marrow but have not yet fully matured.
- Mature neutrophils (mNeu): Mature neutrophils are fully differentiated neutrophils that are capable of phagocytosis and the release of neutrophil extracellular traps (NETs).

In certain diseases, such as cancer, another type of cell related to neutrophils may appear. These cells are called myeloid-derived suppressor cells or MDSCs. They are a type

of myeloid immune cell that has an immunosuppressive nature, with T cells being their primary target [35]. According to recent single-cell studies, this cell type may arise through an abnormal neutrophil maturation process that only occurs in disease cases, giving them their immunosuppressive properties [36].

2.3. Bioinformatic analysis of scRNA-seq

As can be seen, the use of single-cell RNA sequencing (scRNA-seq) has been fundamental in the aforementioned studies. Before the emergence of scRNA-seq, bulk RNA sequencing (Bulk RNAseq) was used, which consisted of processing millions of cells together and mixing their RNA before sequencing. This implied that the result obtained corresponded to an average of the population studied and did not allow the identification of the different cell subpopulations present. The need to analyze these variations was the driving force behind the development of the scRNA-seq technology [37]. This innovative technology enables the detection and quantification of messenger RNAs present in a single cell at high resolution and on a genomic scale. With scRNA-seq, it is possible to describe similarities and differences between transcriptomes within a population of cells, revealing levels of heterogeneity that previously could not be appreciated. In addition, this tool allows the identification and characterization of cell subpopulations present in the analysis, which has transformed the ability to characterize cellular states [38].

The performance of various methods for single-cell RNA-seq analysis has been assessed for tasks such as normalization, feature selection, differential gene expression analysis, clustering, and trajectory analysis [39]. One popular tool for scRNA-seq analysis is Seurat, an R package specifically designed for quality control, analysis, and exploration of single-cell RNA-seq data [40]. Seurat is widely used due to its comprehensive collection of single-cell analysis algorithms and tasks within the R environment. Consequently, the subsequent analysis will primarily focus on utilizing the methods available in Seurat, with particular attention to evaluating the methods employed for normalization, integration, and clustering.

Normalization is a vital process in the analysis of gene expression data, as it aims to adjust for technical differences between samples or characteristics (e.g. genes) that are not biologically significant. In the context of scRNA-seq, normalization is used to mitigate discrepancies in sequencing depth between cells, this is relevant since the differences in gene expression between cells may be due to technical factors rather than biological ones. By applying normalization, it can be ensured that the observed variations in gene expression actually reflect biologically meaningful differences [41]. In the proposed pipelines for Seurat, there are two main normalization methods: LogNormalize [40], which scales by the total number of reads per cell and logarithmically transforms expression measurements, and SCTransform [42], a regularized negative binomial regression model that normalizes and stabilizes data

variance by estimating cell-specific factors. SCTransform is a highly recommended option due to its ability to remove technical artifacts, avoid heuristic steps, and enhance downstream analysis tasks, resulting in more accurate and meaningful results. Compared to LogNormalize, SCTransform accurately models zero counts, reducing technical noise, and improving downstream analysis. By employing a zero-inflated negative binomial model, SCTransform effectively distinguishes between true zeros and dropout events, ensuring reliable normalization and variance stabilization [42]. While there are alternative normalization methods available such as BASiCS (Bayesian analysis of single-cell sequencing data) [43], SCnorm [44] and scran [45], it is important to note that SCTransform also outperforms these methods in terms of removing technical noise, accurately modeling zero counts, and preserving biological heterogeneity [42].

As for clustering, unsupervised clustering corresponds to the process of grouping objects based on similarity without any ground truth or labeled training data. In the context of scRNA-seq, this technique has emerged as a powerful approach to defining cell types based on transcriptome similarity. This is because it is an unbiased, data-driven method that can be applied to any sample. Reliable unsupervised clustering methods are crucial for practical use and represent a key computational challenge in cell atlas construction [46]

The most widely used method for clustering scRNA-seq data corresponds to the Louvain algorithm, this method is also used in Seurat in combination with shared-nearest-neighbor graphs [46]. Unlike other methods that use UMAP or t-SNE reduction in a lower dimensional space, Seurat uses unsupervised clustering with PCA reduction [47]. This approach has the advantage of providing a higher dimensional space for analysis. In addition, Seurat allows users to adjust the "resolution" parameter, which determines the number of clusters, providing flexibility to balance between too many and too few clusters. Thus, Seurat provides a valuable starting point for clustering subpopulations of cells based on their phenotype [39].

Finally, integration is an important step in the analysis of single-cell data, as it corrects for batch effects between samples, allowing the combined analysis of multiple data sets. However, it should be noted that integration is not always necessary and should be evaluated based on the presence of batch effects. If it is necessary, in deciding which integration method to use the main criterion should be to select a method that effectively removes batch effects while maintaining the biological variability present in the data [48]. It is also important to consider practical aspects such as memory usage and run time. In addition, it is crucial to assess whether the integrated results fit with previous biological knowledge and can be validated. This validation step helps to ensure that the chosen integration method is suitable for the specific data set and analysis objectives.

In Seurat, several integration methods are available. One commonly used method is

canonical correlation analysis (CCA) [49] combined with the identification of mutual nearest neighbor (MNN) [50], which allows the identification of shared subpopulations between datasets. Another improved integration method, introduced in 2019, involves the identification of pairwise correspondences between individual cells, known as “anchors”. This method allows datasets to be transformed into a shared space, even in the presence of technical or biological differences [38].

3

Methodology and Materials

The data samples to analyze were provided by Institut Curie. These datasets have already been preprocessed using the Cell Ranger program, which is utilized to align the readings, generate barcode matrices, and perform initial quality control by filtering out empty or multi-cell microspheres.

These processed scRNA sequencing data will be analyzed using R Studio software and using as a basis the Seurat bioinformatics pipeline [51; 52], designed specifically for this type of data, which can be organized in several steps (see Figure 3.1). First, quality control is performed to evaluate the raw data obtained from the single-cell analysis. This involves evaluating sample quality metrics, such as the concentration of mitochondrial RNA present [53]. Poor quality cells are then excluded from further analysis to ensure the reliability of the results. After this filtering, the data undergoes several processing steps, including normalization and scaling, to account for technical variations and improve comparability between samples. Also, genes with higher variability are selected to focus on the most informative features.

Once the data has been processed, linear and non-linear dimensionality reduction techniques, such as principal component analysis (PCA) and uniform manifold approximation and projection (UMAP), are applied to reduce the high-dimensional data into lower-dimensional representations. Then, using clustering algorithms, cells with similar gene expression profiles are grouped, forming distinct clusters that will be identified as a specific myeloid population and saved as a separate file, facilitating further analysis, or removed from the analysis in case they do not correspond to a myeloid population. In this case, the expected populations to identify are dendritic cells, granulocytes, monocytes and macrophages, any other cell type would be excluded before proceeding to the next step.

To identify the corresponding populations and subpopulations, several analyses will be done such as analyzing the known markers of specific populations and subpopulations, also

known as key genes, performing a differential gene expression analysis, analyzing the expression of previously described expression signatures for different cell types in the literature and analyzing the trajectory of maturation and differentiation of the analyzed cells in some cases to confirm their classification. For this last point, packages Monocle2 [54] and Monocle3 [55] will be used instead of Seurat.

Finally, a comparison will be made between subpopulations in different conditions; treated and untreated. This comparison aims to assess the impact of treatment on myeloid subpopulations. For this purpose, the size of each subpopulation in each condition will be determined and the changes will be analyzed using visualization tools such as cell density plots and the numerical data itself. On the other hand, statistical methods, such as the calculation of log2 fold change, will be used to identify significant alterations. Referring to the existing literature, the implications of these changes in the creation of a pro-tumor or anti-tumor environment will be determined. In this way, this analysis will provide information on the dynamic response of myeloid subpopulations and their potential role in the formation of the tumour microenvironment.

It should be noted that, during data analysis, the various methods available in the Seurat package will be carefully studied. If necessary, alternative approaches will be explored based on the guidance of the co-supervising teacher and relevant literature. Flexibility will be exercised in adapting the process to deal with problems that may arise, ensuring that the most appropriate methods are employed to obtain reliable and meaningful results.

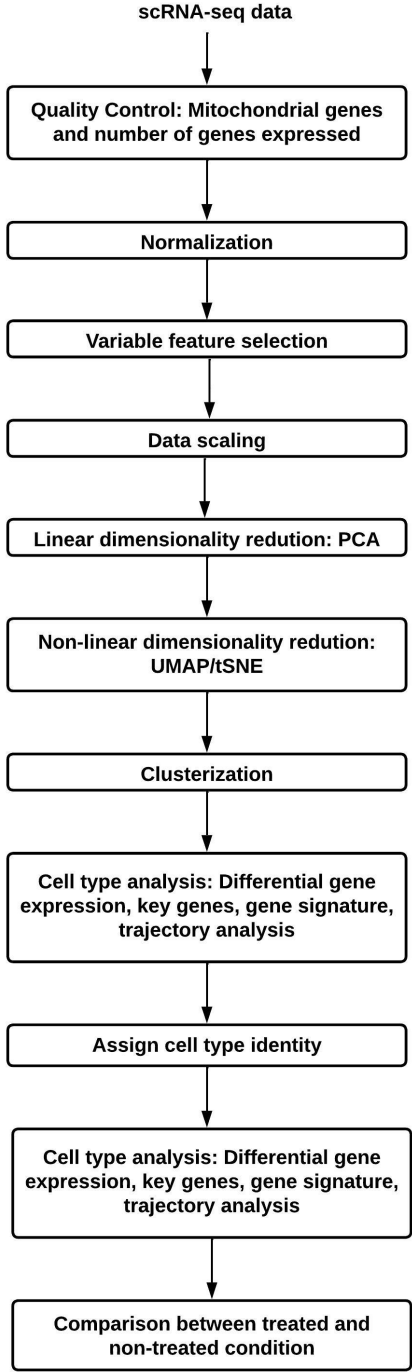


Figure 3.1: scRNA-seq analysis flowchart

4

Results and Discussions

According to a previous study by the Curie Institute, the most abundant type of immune cells in RTs corresponded to monocyte and macrophages [5], followed by granulocytes. Therefore, the study focused on treatments that could affect these population, especially macrophages, since it was observed that a depletion of this population caused a decrease of tumor volume, but systemically depleting macrophages would be too risky. Considering this, several treatments with different agents were analyzed, until it was observed that the immunotherapy treatment to be analyzed in this case was able to reduce tumor growth. Since this treatment was effective it is important to study the effects of it on myeloid populations to comprehend why this happens. For more information on these results, please refer to Appendix A.

The methods utilized for the analysis of the scRNA-seq datasets correspond to the ones that can be seen in figure 4.1. This corresponds to an adapted version of the typical Seurat pipeline, starting from the quality control of the separated datasets and ending with the separation of each myeloid population. A detailed explanation and the code used for this pipeline can be found in appendix B.

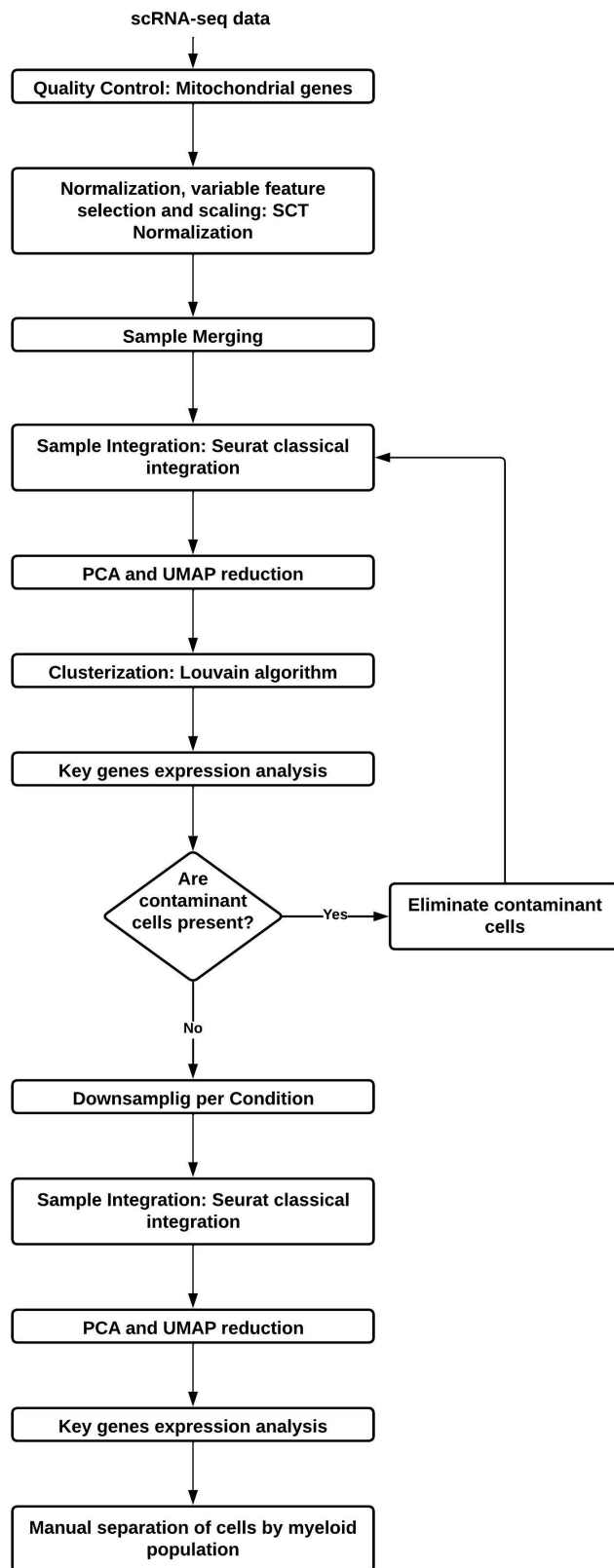


Figure 4.1: General data pipeline

Usually, during quality control, cells with high mitochondrial gene expression and a low number of genes expressed (nFeatures), are filtered out due to these being signs of cell damage or low cell quality [53]. However, in this case, as granulocytes are included in the analysis, the nFeature filter cannot be applied since these cell types, particularly neutrophils, tend to exhibit a lower number of expressed features. Therefore, quality control was conducted exclusively considering the percentage of mitochondrial genes present. Each dataset's content of mitochondrial genes was analyzed using scatter plots, both with the raw data and logarithmic values (see figures 4.2 to 4.5).

Untreated sample 1:

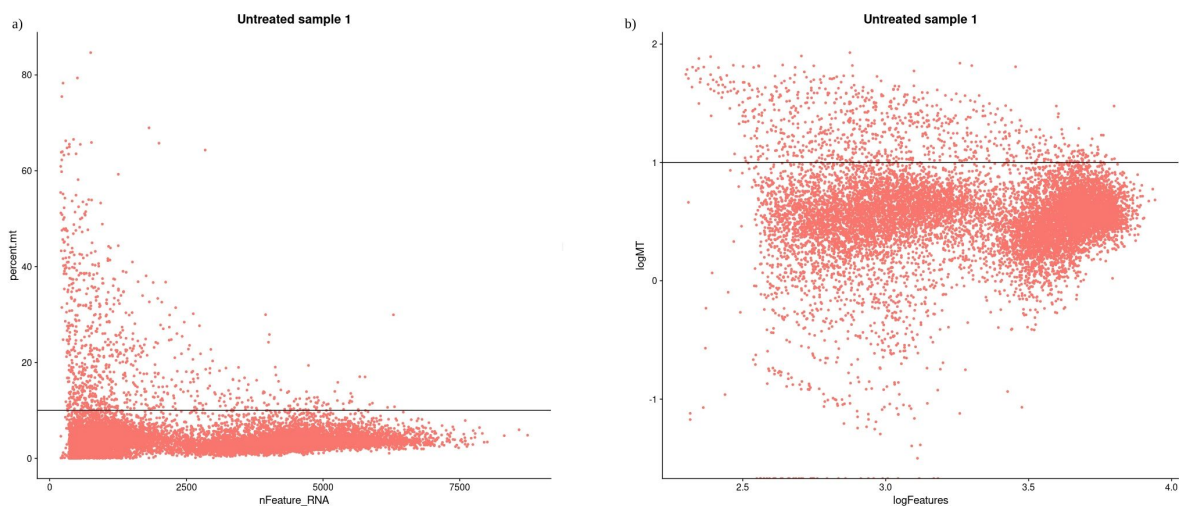


Figure 4.2: Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

Untreated sample 2:

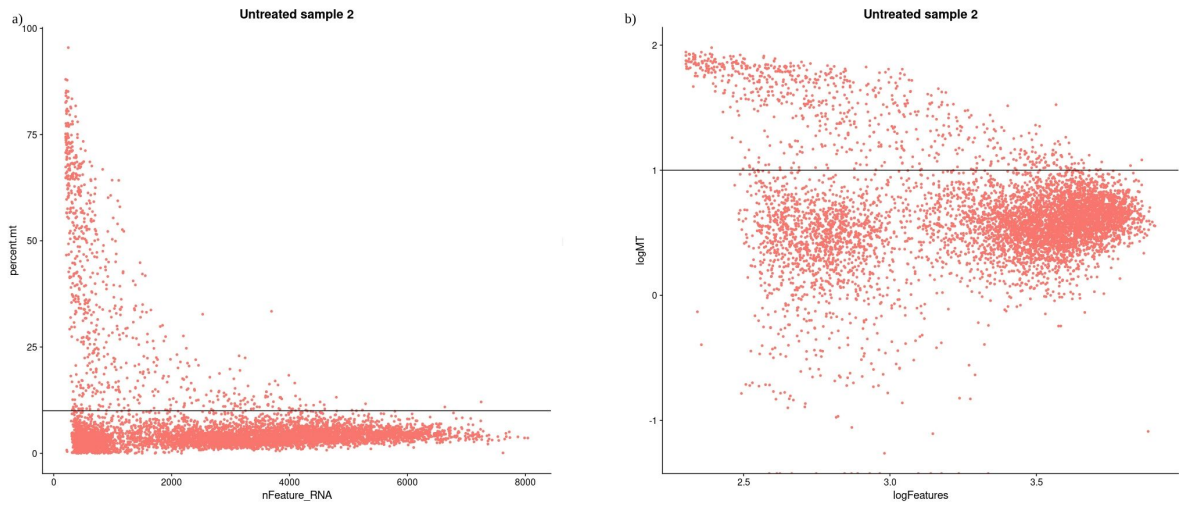


Figure 4.3: Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

Treated sample 1:

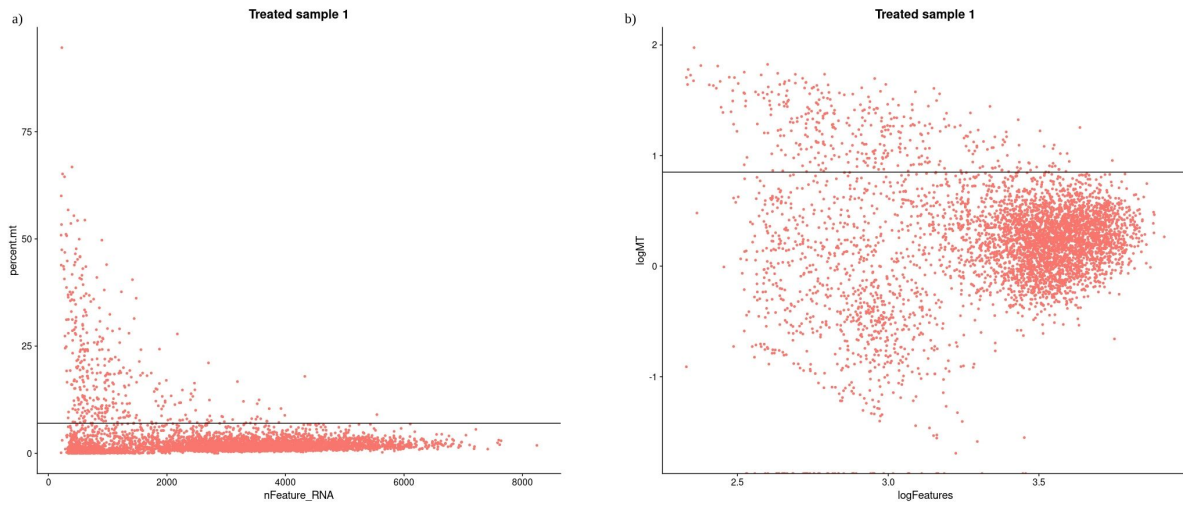


Figure 4.4: Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

Treated sample 2:

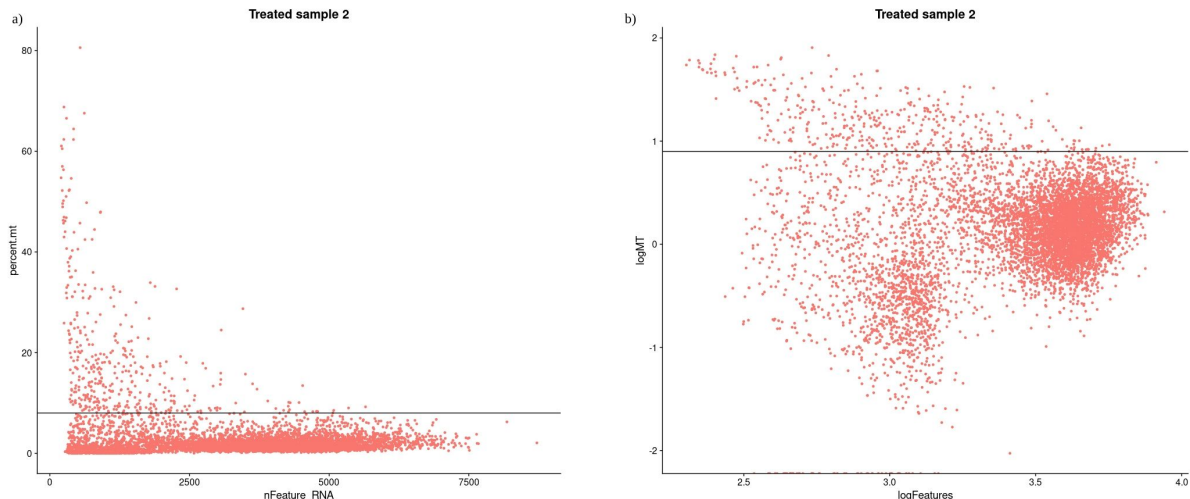


Figure 4.5: Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

In this case, different cut-off points were utilized for each sample. For Untreated samples 1 and 2, here forward mentioned as Untreated1 and Untreated2, both had a cut-off point of 10% mitochondrial gene content, i.e. all cells with a higher than 10% content of mitochondrial genes were eliminated. In the case of Treated samples 1 and 2, here forward mentioned as Treated1 and Treated2, cut-off points were 7% and 8 % respectively. The general strategy to determine the cut-off point for each of the four samples was to cut above the principal cumulus of cells that can be seen in the logarithmic values Scatter plots. In the case of Untreated samples, this cumulus had a similar vertical distribution, resulting in both samples being cut at the same percentage, while Treated samples generally were distributed in a lower vertical position, resulting in lower cut-off points. These differences, more than being the result of the applied treatment, are likely generated mostly due to batch effect in the process of obtaining the data, so it is not an important difference to consider in the analysis of the effects of the Treatment.

After quality control comes the normalization process. Each sample was normalized individually, using SCTransform. Initially, the number of variable features considered for the normalization was 3000, as this is the default value that is normally used. However, in future analyses, this number was increased to 5000 so that more genes would be considered. After the normalization, the samples were merged to determine if it would be necessary to integrate them. To visualize the result of the merging samples, both Principal Component Analysis (PCA) and UMAP dimensionality reductions were used. To determine the number of principal components (PCs) to use for UMAP reduction, an Elbow plot was utilized. Based on the plot, it was decided to use 40 PCs to generate the merged UMAP visualization (see Figure

4.6)

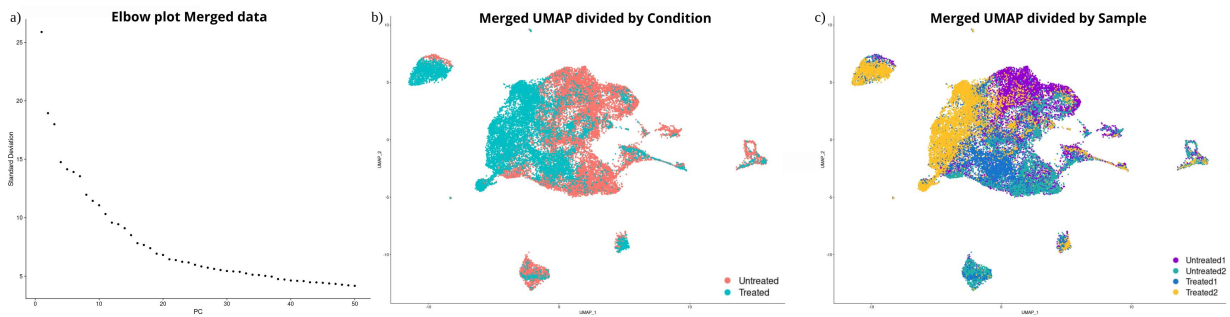


Figure 4.6: Merged data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples.

In these figures, we can see that in some regions there are cells mostly from only one specific sample, therefore, it is considered necessary to perform an integration of the samples. For the integration, the method used was the classical Seurat integration (CCA + MNN). After integration, in this case, according to the Elbow plot, 35 PC's were used to generate the integrated data UMAP (see Figure 4.7).

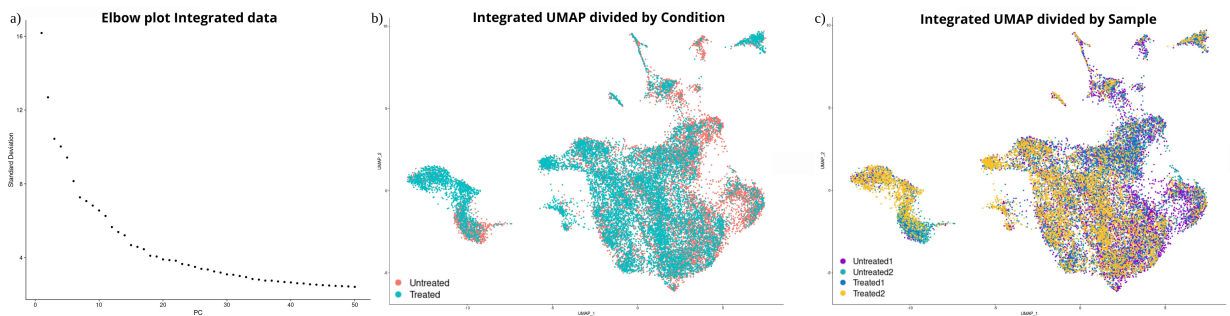


Figure 4.7: Integrated data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples.

In this new UMAP we don't have groups of cells corresponding to only one sample, therefore we can proceed with the analysis. The next step is to make sure that only myeloid cells are present in the data, for this, we have to eliminate other contaminant cells. First, the cells are divided into groups, or clusters, according to the similarity between them, using the Louvain algorithm. To choose an appropriate resolution, the parameter which determines the number of clusters formed, a Clustree graphic was used. In this case, the chosen resolution was 0.1 (see Figure 4.8).

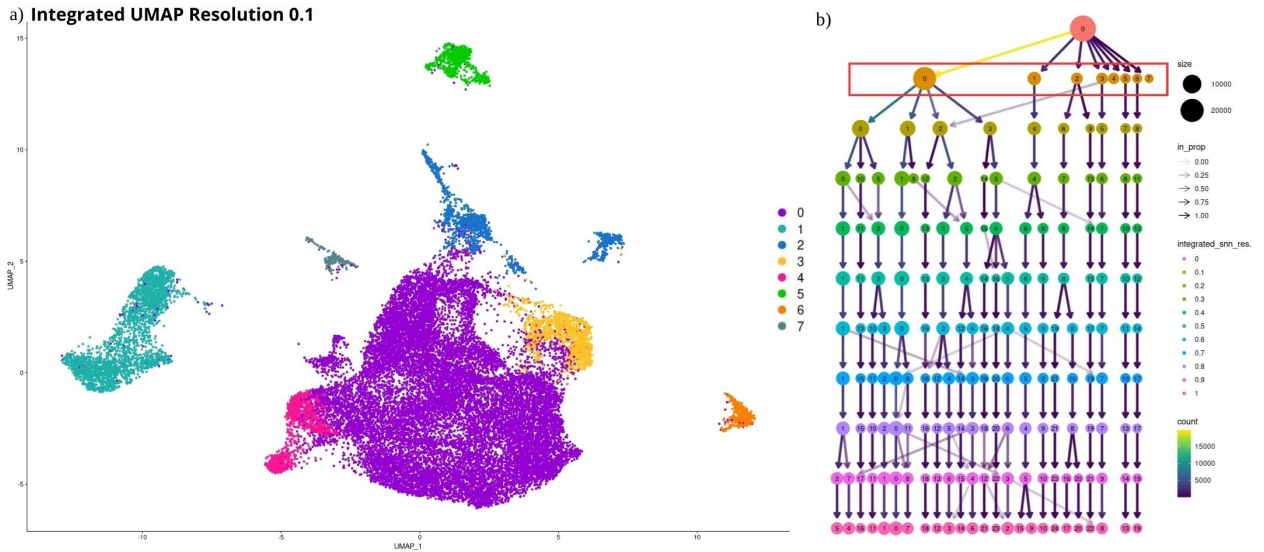


Figure 4.8: Integrated data clustering. a) Integrated data UMAP resolution 0.1. b) Integrated data Clustree.

Now that the cells are grouped, we look for specific marker genes to identify contaminants. One of these possible contaminant populations are T cells, this group can be identified with the marker gene *Cd3e* (see Figure 4.9). We can see that cluster 5 has a high expression of gene *Cd3e* and is a separate group from other cells. This cluster was determined to be contaminant T cells and was eliminated from the data. After this, the remaining cells were re-integrated.

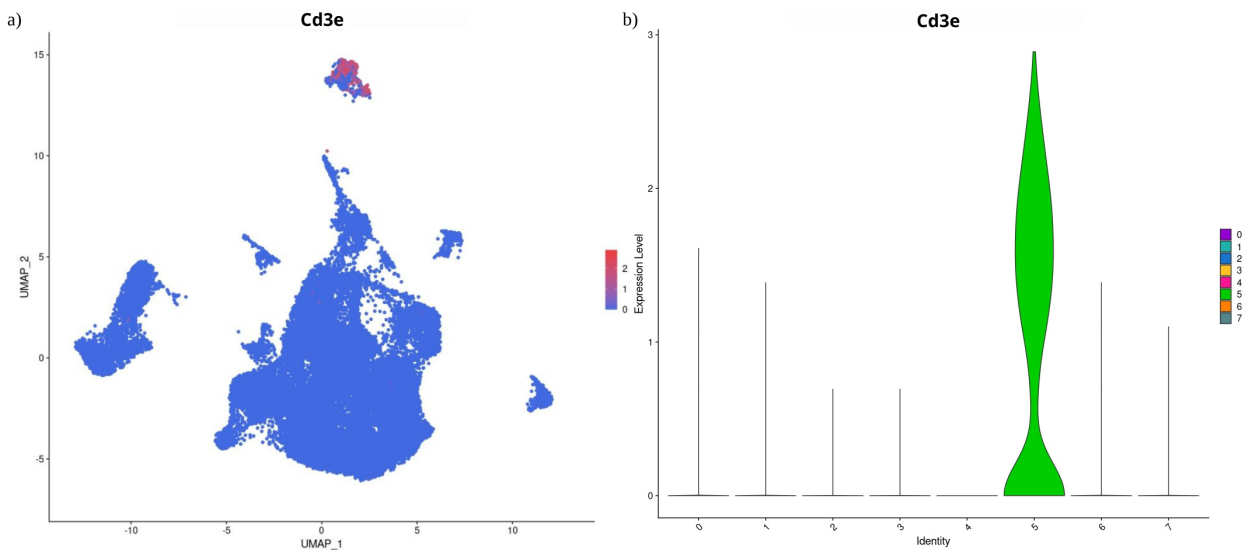


Figure 4.9: *Cd3e* analysis. a) *Cd3e* expression level. b) *Cd3e* violin plot.

This process of identifying contaminant cells, eliminating them and re-integrating the

remaining cells was repeated until it was considered that there were no contaminant cells. For this, the principal gene analyzed was *Ptpnc*, as this is a common myeloid and lymphoid cells gene marker. Therefore, genes with low or negative expression of *Ptpnc* were eliminated. After this process, to ensure the equal representation of each condition in the analysis, a downsampling process was performed so we had the same number of treated and untreated cells, this being a total of 10968 cells per condition. For this final re-integration, the number of genes used for normalization was augmented to 5000 genes and the number of PC's used was 50. This generated the new and final UMAP of myeloid cells (see Figure 4.10)

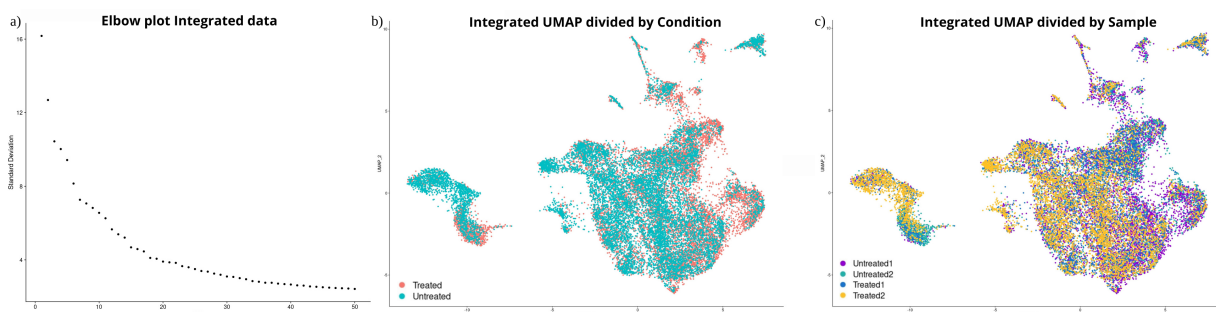


Figure 4.10: Integrated data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples.

With this final UMAP, we analyze the expression of each myeloid population's key genes: *Mmp9*, *Csf1r* and *Flt3*, to identify the populations of granulocytes, monocyte/macrophages and dendritic cells, respectively. According to the expression of these three genes the cells were manually separated in their respective populations (see Figure 4.11).

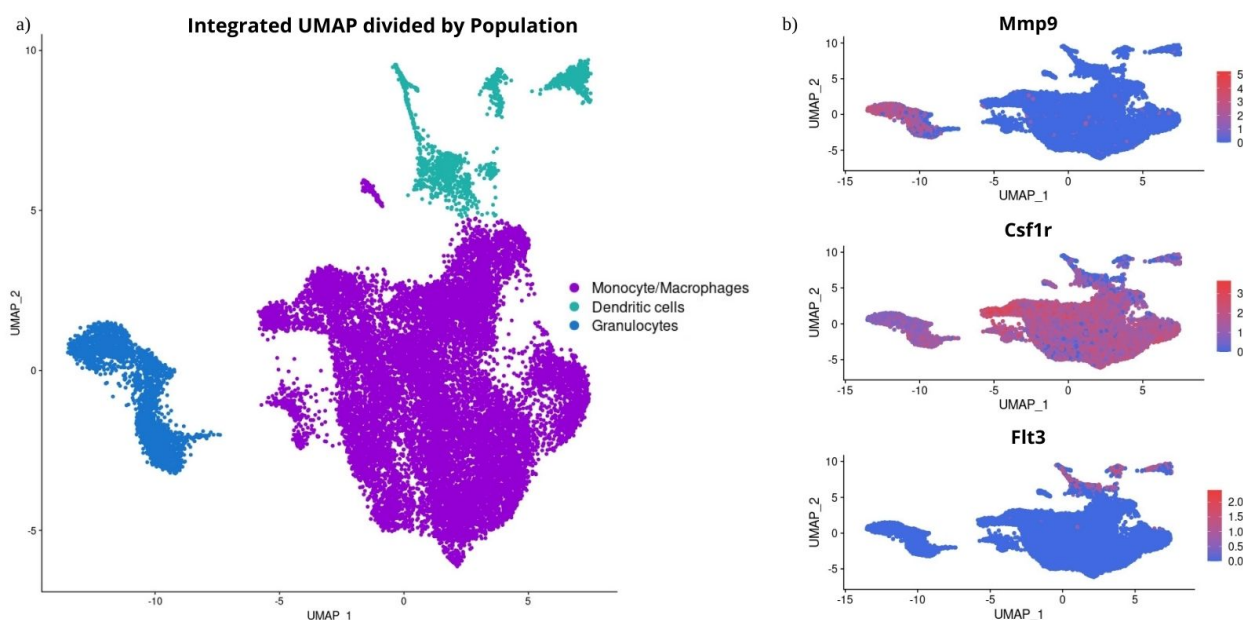


Figure 4.11: Myeloid cells classification. a) UMAP divided by populations. b) Key genes expression analysis: *Mmp9*, *Csf1r* and *Flt3*

To ensure that this separation was done correctly, the expression of other key genes of myeloid populations was analyzed using a Dotplot graphic (see Figure 4.12). This analysis confirmed the validity of the separation.

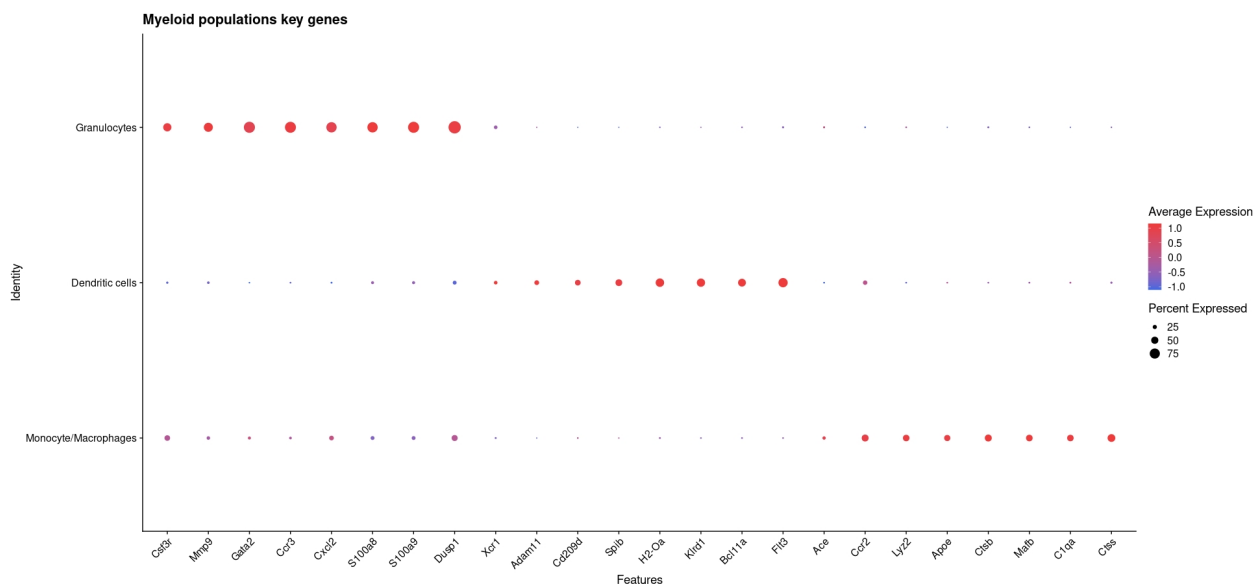


Figure 4.12: Myeloid key genes Dotplot

With this initial result, it can already be seen that there are quantitative differences between conditions. Under the treatment condition, the granulocyte proportion increased, whereas both dendritic cells and monocyte/macrophage proportions decreased (see Figure 4.13). To quantify these changes, the value of their log₂ Fold Change (log₂FC) was calculated (see Appendix F), the results obtained were a log₂FC of -0.05 for monocyte/macrophages, -0.67 for dendritic cells and 0.74 for granulocytes.

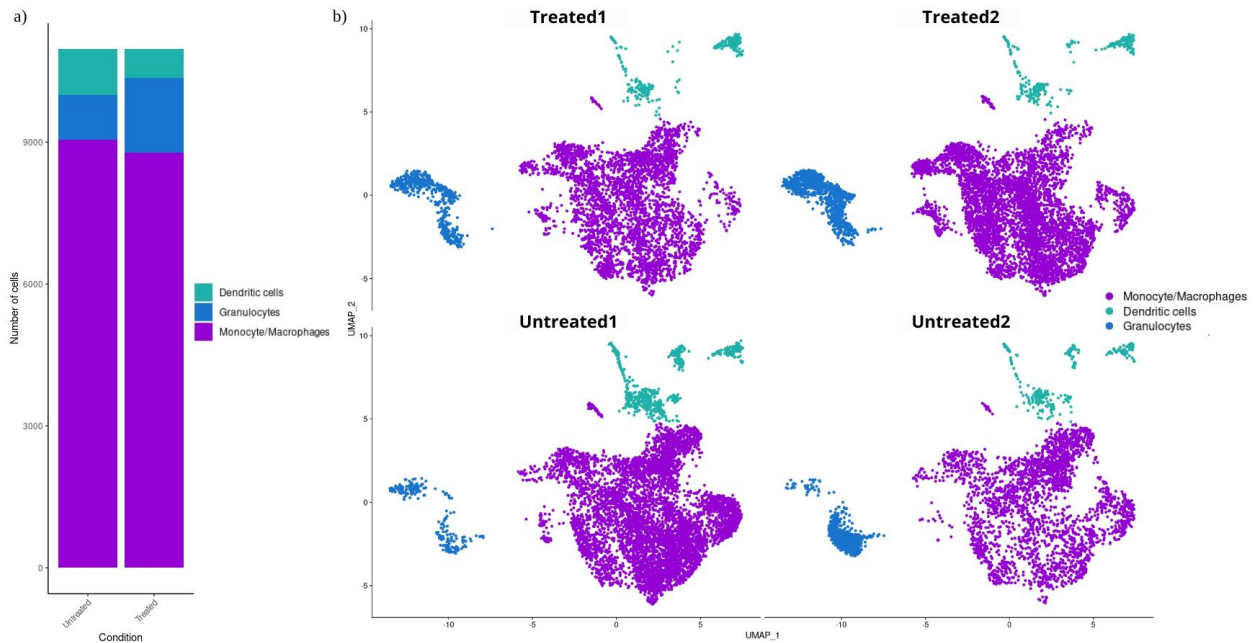


Figure 4.13: Myeloid populations changes analysis. a) Barplot of populations percentage divided by Condition. b) Myeloid populations UMAPs divided by Sample.

These results indicate that only the granulocyte population increased. This is important information to have in consideration, however, to fully comprehend the impact of the treatment and the consequential qualitative changes, each population must be studied separately. This way, the differences between the subpopulations that compose the populations in each condition can be analyzed in more detail. Due to the different nature of each population, they were all separated in the following steps so they don't interfere with the analysis of the other. It is important to note that, for both monocyte/macrophage population and dendritic cells, a new quality control process was applied, since now, without the presence of granulocytes, a feature expression filter could be applied. In both cases, the cut-off point was established at 2.000, meaning that cells expressing fewer than 2.000 genes were eliminated.

4.1. Dendritic cells

The specific methods used for dendritic cell analysis is depicted in Figure 4.14, while the detailed code used for it can be found in Appendix C, in which process of the second quality control is also included.

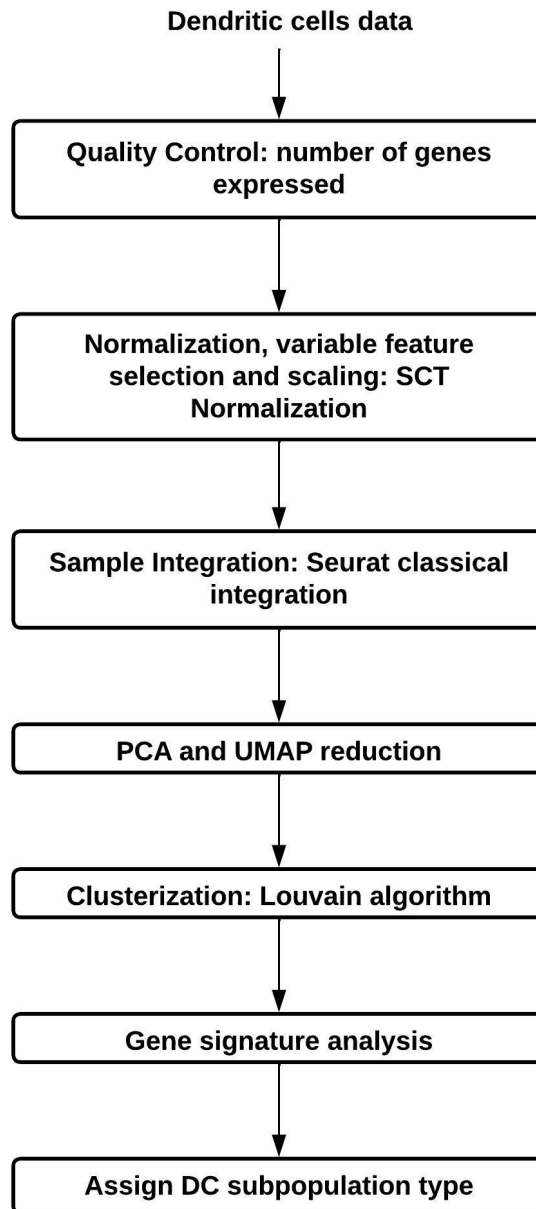


Figure 4.14: Dendritic cells pipeline

Just as it was done in the previous section, the dendritic cell data was integrated by

sample. Then, for UMAP reduction a total of 40 PC's were used, generating the graphics on Figure 4.15.

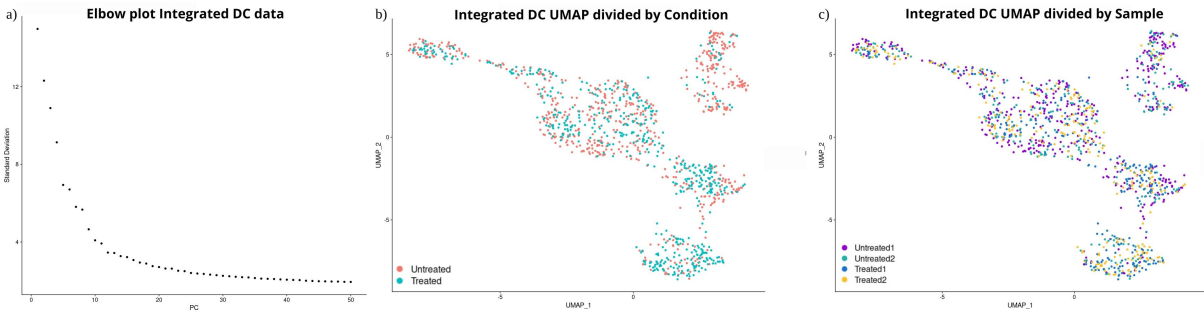


Figure 4.15: Dendritic cells data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples.

Next, for the clustering of the cells, the chosen resolution was 0.2, this was done by analyzing the corresponding Clustree (see Figure 4.16), in which it is important to notice the stability in the cluster membership between resolutions. This stability establishes that there is a clear division between the different DC subpopulations.

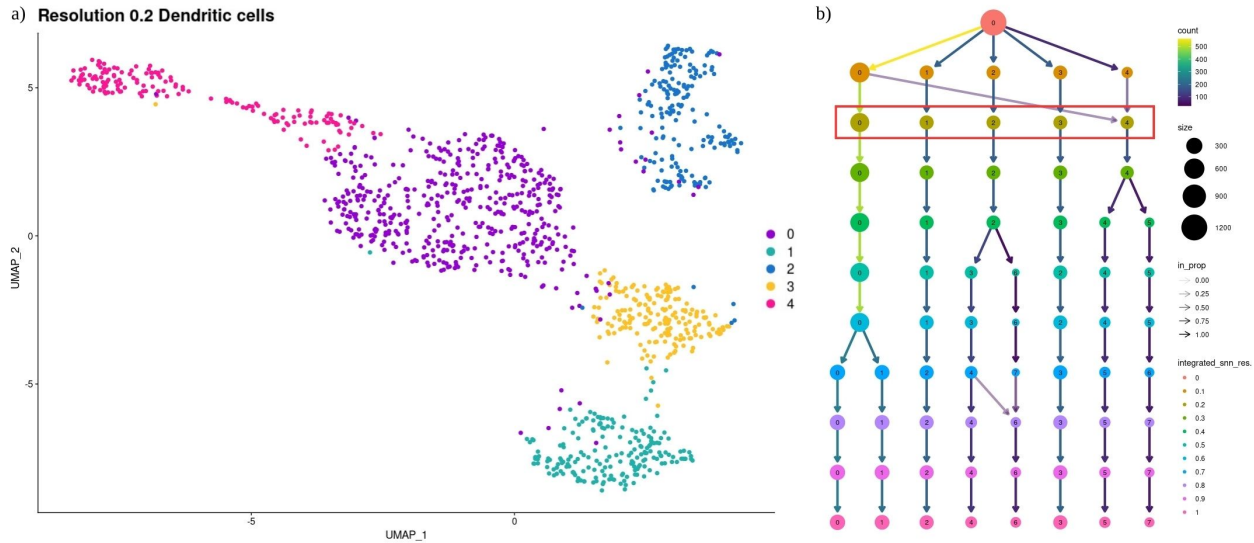


Figure 4.16: Dendritic cells clustering. a) Dendritic cells UMAP resolution 0.2. b) Dendritic cells Clustree.

Regarding the identification of the different subpopulations, as mentioned before, the division in dendritic cells subpopulations was thought to be a very established one, however, recently there have been some issues with this, especially regarding subpopulation DC3. There are currently two branches of DC classification using the name DC3 for two different subsets of cells. On one hand, we have a DC3 population that is aligned with signatures, con-

sisting of sets of most relevant genes, for mregDC and Ccr7+ subsets [29; 31; 33], which makes this particular DC subpopulation characterized by a mature, active and immunoregulatory profile. On the other hand, we have a DC3 subpopulation that will replace the previously known MoDC subpopulation, as this DC3 subpopulation corresponds to a distinct lineage of dendritic cells that arises from Ly6C+ monocyte-dendritic cell progenitors. As this subset has been identified as an independent lineage of dendritic cells, not arising from the same lineage of cDCs, it would replace the traditional MoDC classification [30]. It is important to note that these DC3 subsets are not the same, as they have two distinct gene signatures and functions. The second type of DC3 subset has a more active role in T helper cell recruitment, without any noticeable immunoregulation function. According to this classification, there is another subset with immunoregulation functions, referred to as mregDC, that coincides with the DC3 subset of the first branch of classification. Since this is a problem related mostly to the name used to categorize dendritic cells, it was decided to use the names of the branch that has been in use for a longer time; the immunoregulatory DC3 branch.

With this in consideration, gene signatures, which consist of sets of genes, were utilized to identify the various subpopulations present in the data (see Figure 4.17). In total 5 different populations were identified; cDC1 [21; 25–27; 29–33; 56], cDC2 [21; 25–33; 56], MoDC [28; 29; 31; 33], pDC [21; 25–27; 29; 30; 32; 33] and DC3 [21; 27; 29–33].

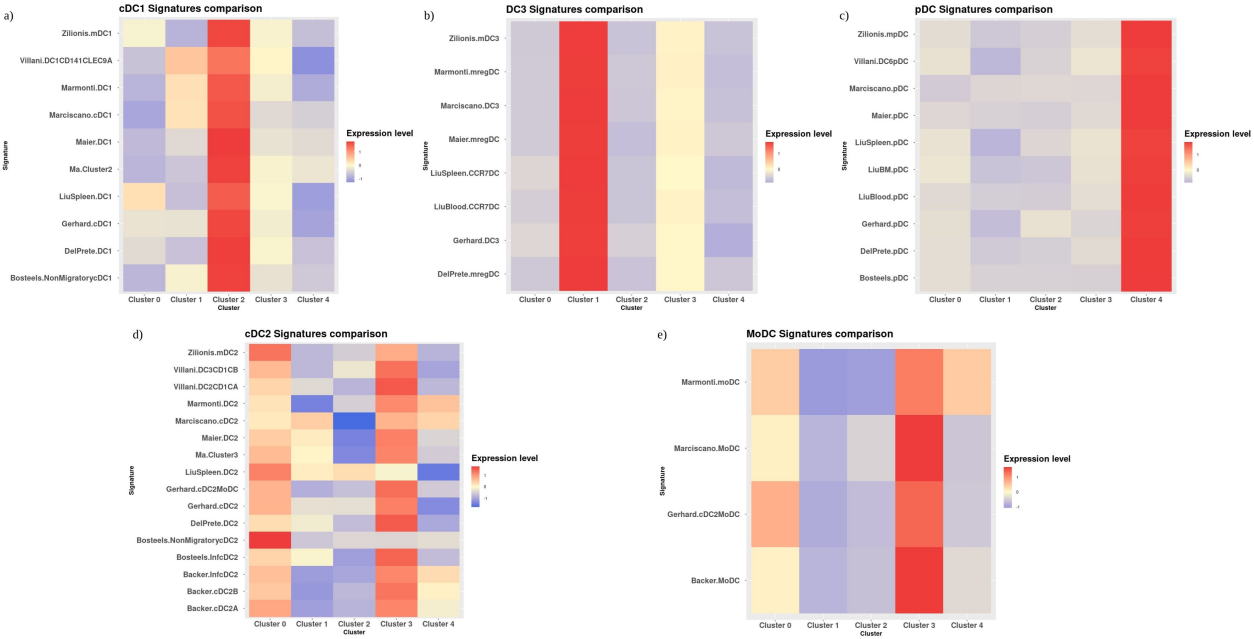


Figure 4.17: Dendritic cells Heatmaps. a) cDC1 signatures. b) DC3 signature. c) pDC signatures. d) cDC2 signatures. e) MoDC signatures.

Regarding this classification, one thing to notice is that the cDC2 subpopulation was separated from MoDC (see Figure 4.18). In this case, both clusters 0 and 3 exhibited high

levels of expression for cDC2 and MoDC signatures. However, in the case of cluster 3 MoDC signatures seemed to be more predominant, therefore this cluster was classified as a MoDC cluster, even though it could also be considered as cDC2/MoDC. On the other hand, cluster 0 did not exhibit enough expression of MoDC signatures compared to the previous group and was, therefore, only considered a cDC2 cluster.

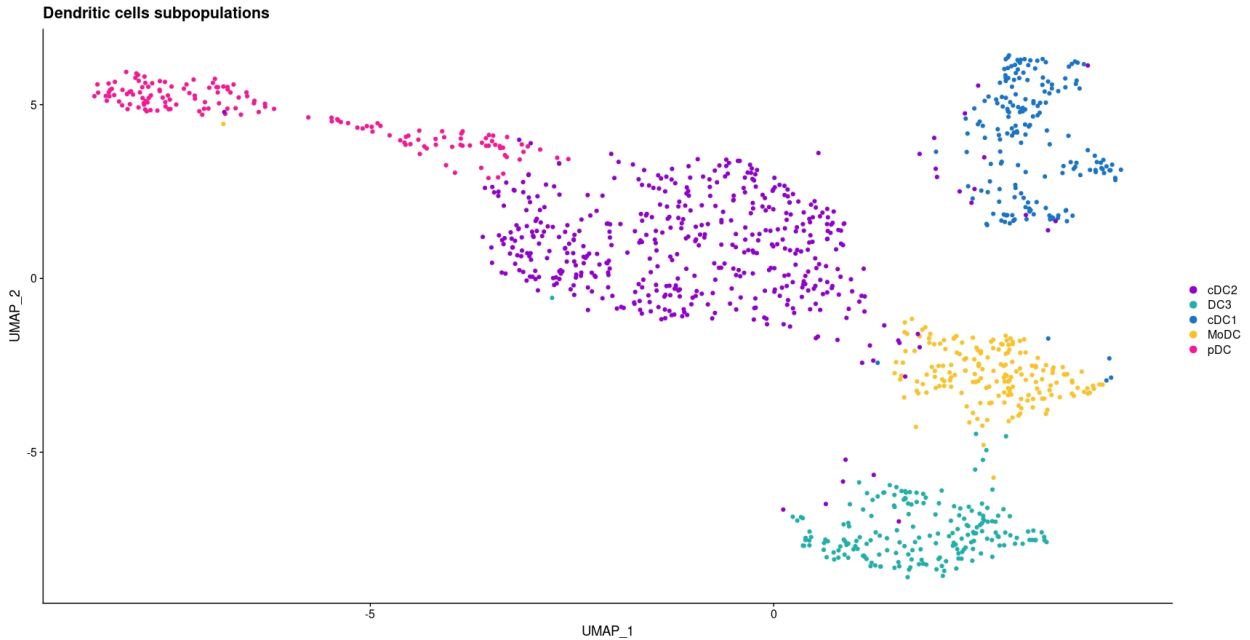


Figure 4.18: Dendritic cells subpopulations.

In regards to quantitative changes in subpopulations, in the Treated condition there was an increase of MoDC and CD3 populations (see Figure 4.19), of 0.25 and 1.20 log2FC respectively, while the other three subpopulations decreased (see Appendix F.2).

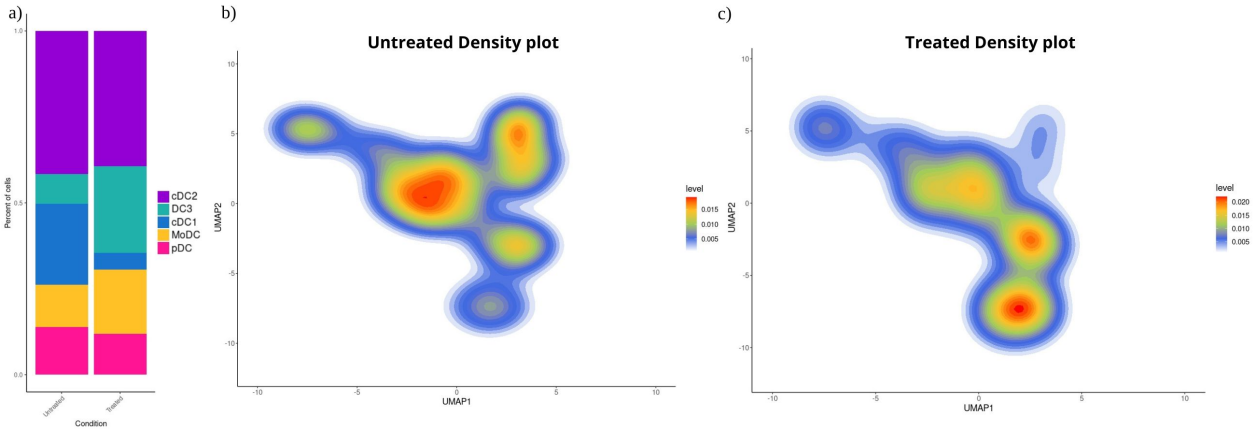


Figure 4.19: Dendritic cells subpopulations changes. a) Barplot of dendritic cells subpopulations percent by Conditions. b) Untreated dendritic cells density plots. c) Treated dendritic cells density plots

In terms of qualitative changes, both cDC1 and cDC2 have the ability to recruit T cells [26; 27], they do this by migrating to the lymph nodes, where they engage in antigen presentation and T cell activation [27], this migration might explain the decrease in the cDC population in the tumor. The recruitment of more T cells into the tumor microenvironment would have a positive effect in terms of response to treatment since these cells are the primary focus in several immunotherapeutic studies against tumors due to their anti-tumor activity [5]. On the other hand, DC3 also have the ability to migrate to the lymph nodes, due to their high expression of Ccr7, but they can also remain within the tumor [21], which might be what is happening in these cases, since that would explain why their population doesn't decrease like cDC1 and cDC2. The increase in DC3 population is relevant regarding the effects of the therapy because these cells express immunostimulatory molecules, that could both activate CD8+ T cells [27; 29; 31; 33] and CD4+ T cells [31] responses. However, they also express immunoregulatory genes such as PD-L1 and IL4 (down regulator of interleukin-12 (IL-12) production) [27; 31–33]. The expression of these genes can have a negative effect on the response to the treatment, exhausting T cells [33] and contributing to immune suppression [27]. Considering these, for a better response using this treatment it would be good to combine it with immune checkpoint blockade treatments such as anti-PD1, which has had good results in Rhabdoid Tumors treatment [5] so the recruited T cell, both by cDCs and DC3s, do not suffer from cell exhaustion.

4.2. Monocyte and Macrophages

As was done for dendritic cells, the monocyte/macrophages data was also subjected to a second quality control based on the number of genes or features expressed. In this case, the cut-off point was also established to be 2000 genes. The details regarding the selected methods and the whole pipeline used for the analysis can be found in appendix D. The most relevant divergence from the previously used pipeline is that, due to the high plasticity of these cells, instead of using specific signatures a differential gene expression analysis was performed, on this basis a key gene analysis and a trajectory analysis were also done to complement the information obtained (see Figure 4.20).

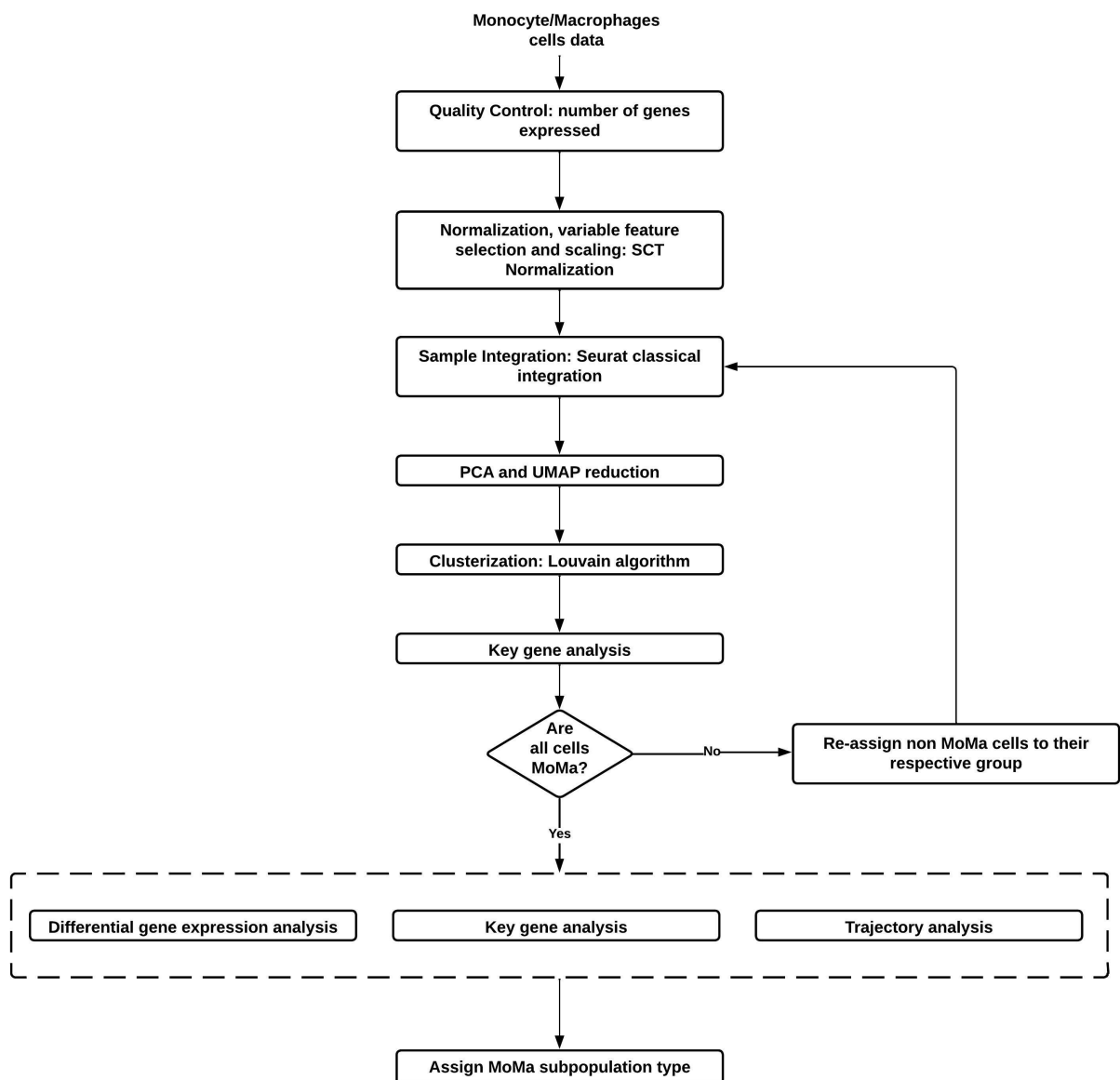


Figure 4.20: Monocyte/Macrophages pipeline

After the integration process, the PCA and UMAP reductions were performed, using 40 PC's (see Figure 4.21).

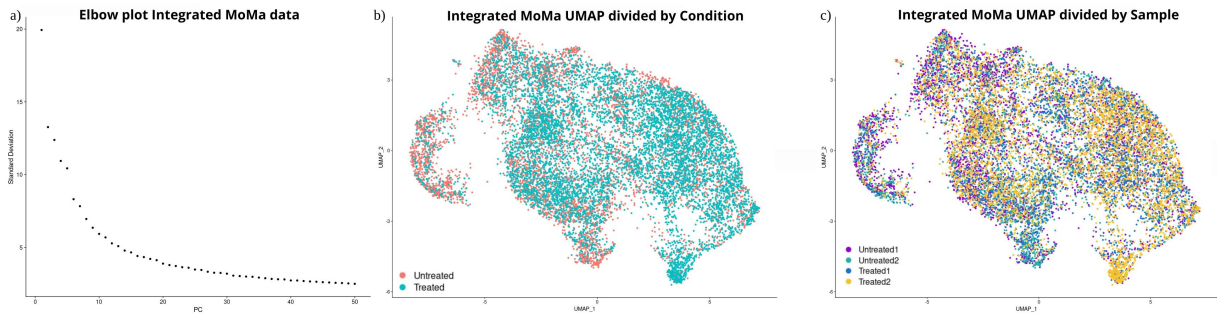


Figure 4.21: Monocyte/Macrophages data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples.

Contrary to what was observed in dendritic cells, the monocyte and macrophages clustree is constantly dividing between each resolution and interchanging cells between clusters. In this case, the resolution that was considered to be the most stable and represented better the division of the cells was resolution 0.7 (see Figure 4.22)

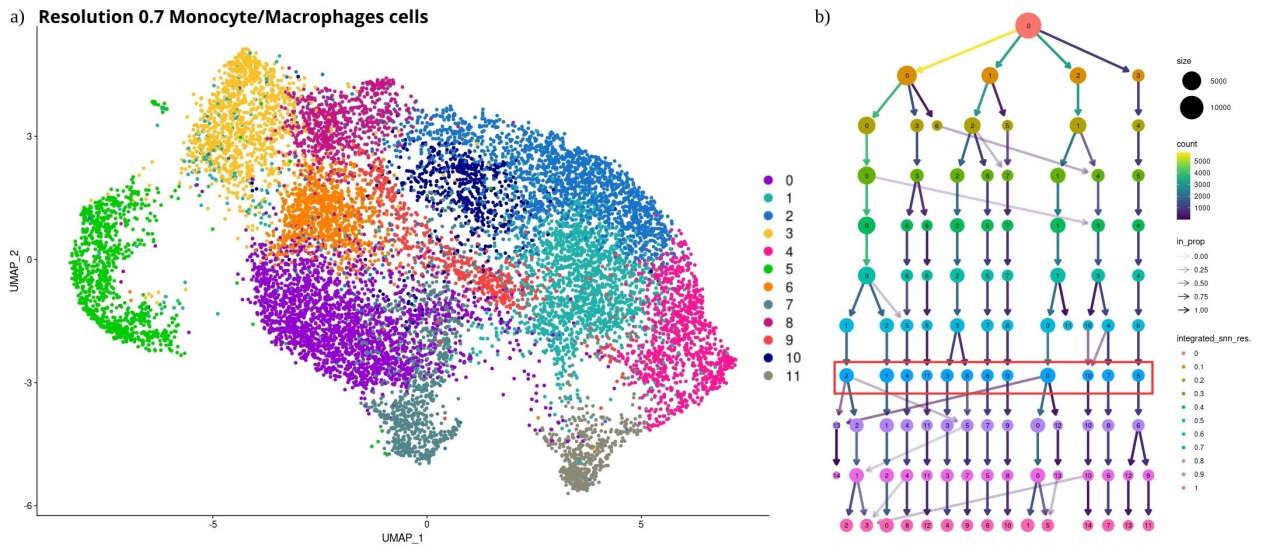


Figure 4.22: Monocyte/Macrophages clustering. a) Monocyte/Macrophages UMAP resolution 0.2. b) Monocyte/Macrophages Clustree.

At this resolution, to determine the most representative genes of each cluster, a differential gene expression analysis was performed (see Figure 4.23). Based on this data, some genes were selected to test their gene expression level, alongside other key genes such as *Csf1r*, *Ly6c2*, corresponding to classical monocyte marker Ly6C [13; 21]; *Spn* (CD43), for resident monocytes [13]; histocompatibility type II gene *H2-Aa* and *ApoE* for macrophages;

Ifit1 for interferon response macrophages (IFN); Mki67 for cycling macrophages; Hmox1 for Mregs; Itgax and Flt3 for monocyte derived cells (see Figure 4.24).

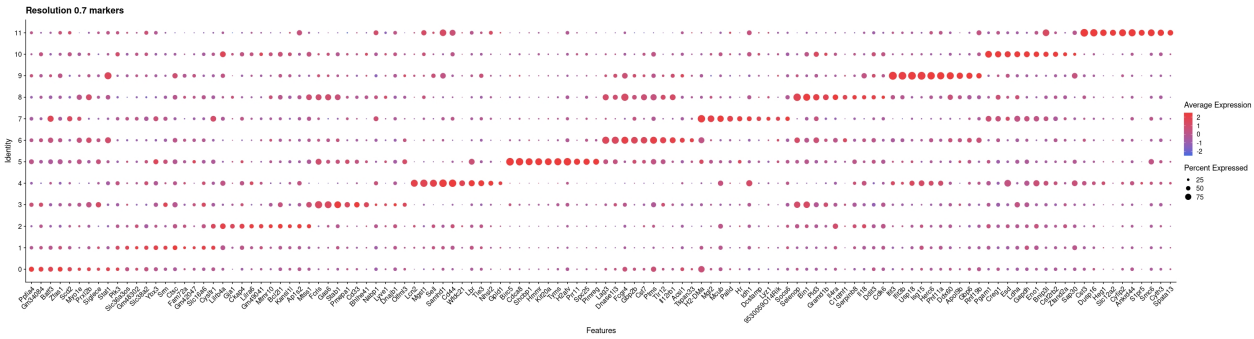


Figure 4.23: Monocyte/Macrophages resolution 0.7 Dotplot

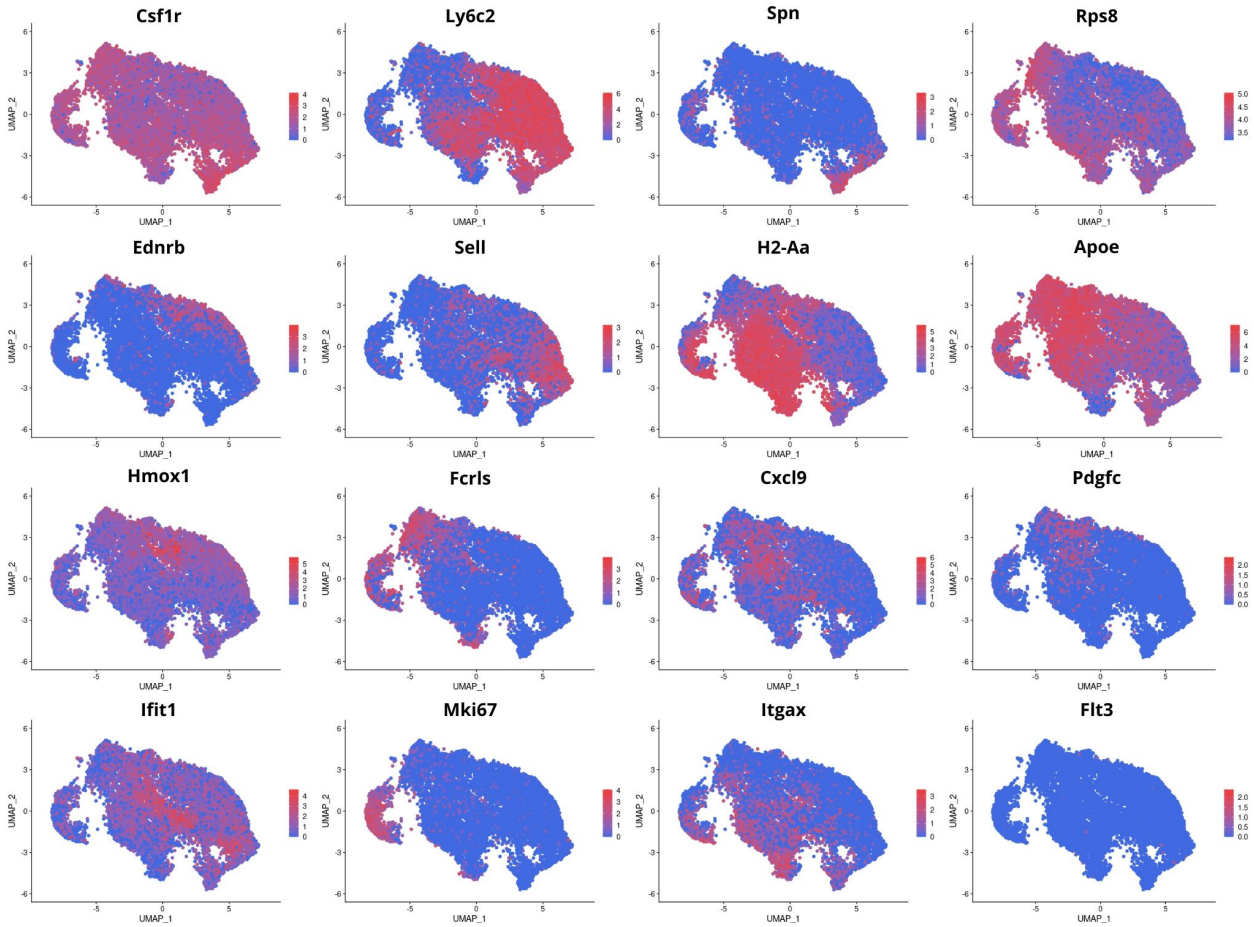


Figure 4.24: Key genes expression analysis

According to these gene expression patterns, monocytes subpopulations were identified to be on right side of the cell group, while macrophages are on the left. On the bottom part of the middle section, we have monocyte-derived DC (Flt3-) and monocyte-derived macrophages,

which are monocyte cells that are differentiating into dendritic cells and macrophages respectively. To prove this hypothesis Monocle 3 was used to generate a trajectory (see Figure 4.25). With the pseudotime the maturation level of each cell can be evaluated since monocytes are considered to be less mature than macrophages, and using the trajectory itself we can see if the clusters named as monocyte-derived follow indeed the path to differentiating into the other cells. In general, the hypothesis was proved, as we can see how the pseudotime increases from right to left and we see a division at the bottom middle part, as some monocytes mature into macrophages, going up to the left side, while others take a monocyte-derive DC profile going further down.

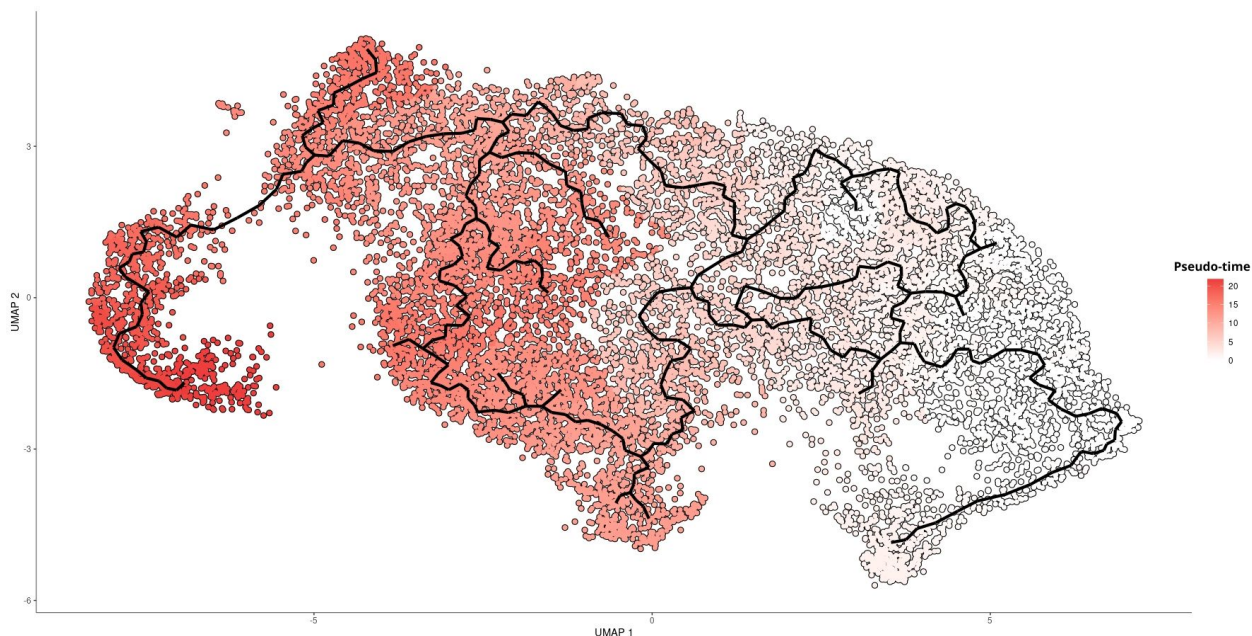


Figure 4.25: Monocyte/Macrophages trajectory using Monocle3

It's worth mentioning that Monocle2 was originally intended to be used for trajectory analysis because Monocle3 requires manual assignment of an initial cluster to generate pseudotime, whereas Monocle2 does it automatically and therefore gives a less biased result. However, the computational cost for Monocle2 turned out to be much higher than expected, as R Studio would crash each time it was attempted to create a trajectory with it. Therefore, Monocle3 was used instead. Despite this, the results obtained with Monocle3 are consistent with the genes studied, and as such, it is considered that the obtained results with are sufficient to categorize the clusters.

Based on the information obtained with these analyses, the subpopulations identities were assigned. Firstly, they were divided in monocyte, macrophages and monocyte-derived cells, and then more specific details were added to each cluster. For more established groups such as resident monocytes, Mregs, cycling and IFN macrophages these classifications were used,

while other clusters were identified by their most relevant and differentially expressed genes, like Cxcl9 macrophages or Sell monocyte (see Figure 4.26).

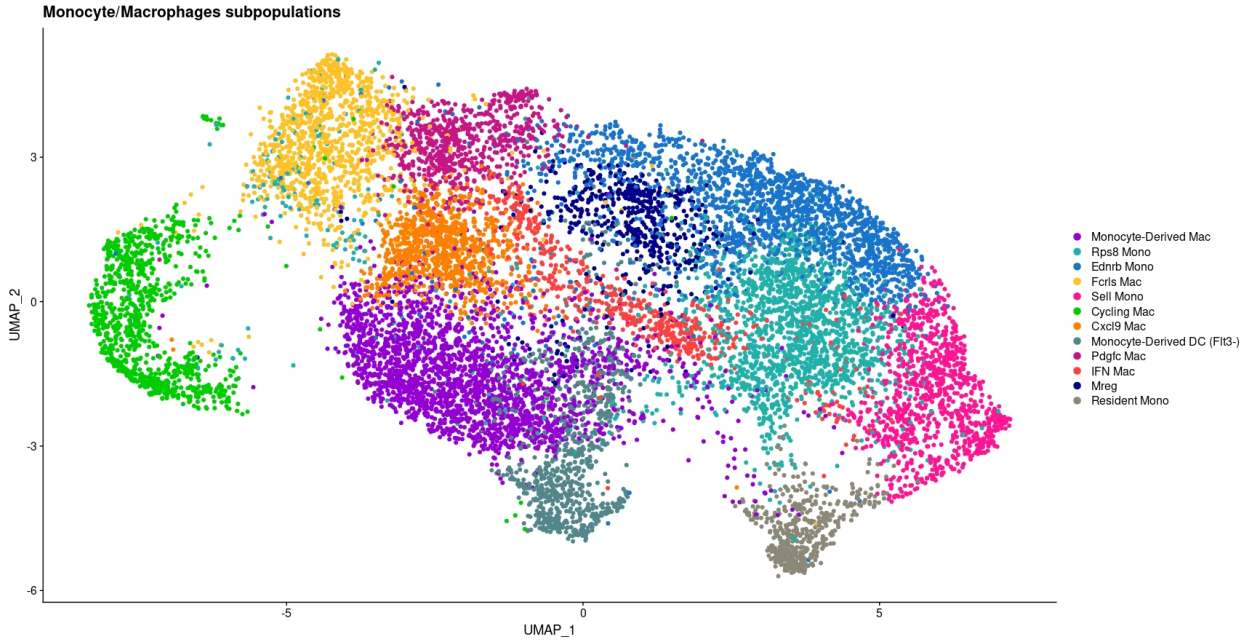


Figure 4.26: Monocyte/Macrophages subpopulations

In terms of the quantitative changes observed between condition, the most substantial difference is that under treatment condition monocyte populations seem to increase (see Figure 4.27), with the most increase being that of subpopulation Resident monocytes, with a log2FC of 2.20 (see Appendix F.3). However, the most interesting change, in terms of qualitative effects, is the increase of macrophage subpopulation “Cxcl9 Mac”.

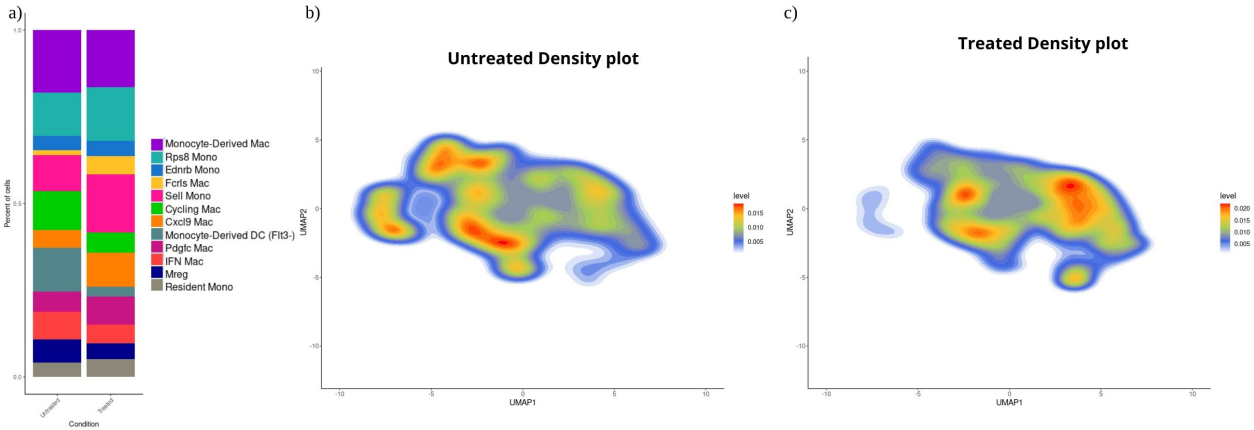


Figure 4.27: Monocyte/Macrophages subpopulations changes. a) Barplot of monocyte/macrophages subpopulations percent by Conditions. b) Untreated monocyte/macrophages density plots. c) Treated monocyte/macrophages density plots

Cxcl9 currently is considered a marker gene for macrophage polarization, leaning towards a more anti-tumoral phenotype of TAMs, it has even been observed that patients with a higher expression level of CXCL9 have better clinical outcomes [22]. Considering this, macrophages were analyzed by themselves to determine if this phenomenon was something exclusive to this specific cluster or if Cxcl9 expression increased in general with this treatment. Spp1 expression was also tested, as this gene is supposed to be the polar opposite of Cxcl9, giving a more pro-tumoral phenotype to TAMs[22]. According to this analysis, Spp1 expression decreased under treatment condition, while Cxcl9 increased through the macrophage clusters (see Figure 4.28), showing a qualitative change towards an anti-tumoral phenotype. Not only this but since Cxcl9 anti-tumour role comes from its function as a chemokine, this being the activation and recruitment of T cells [22], other chemokine gene was also tested; Cxcl10.

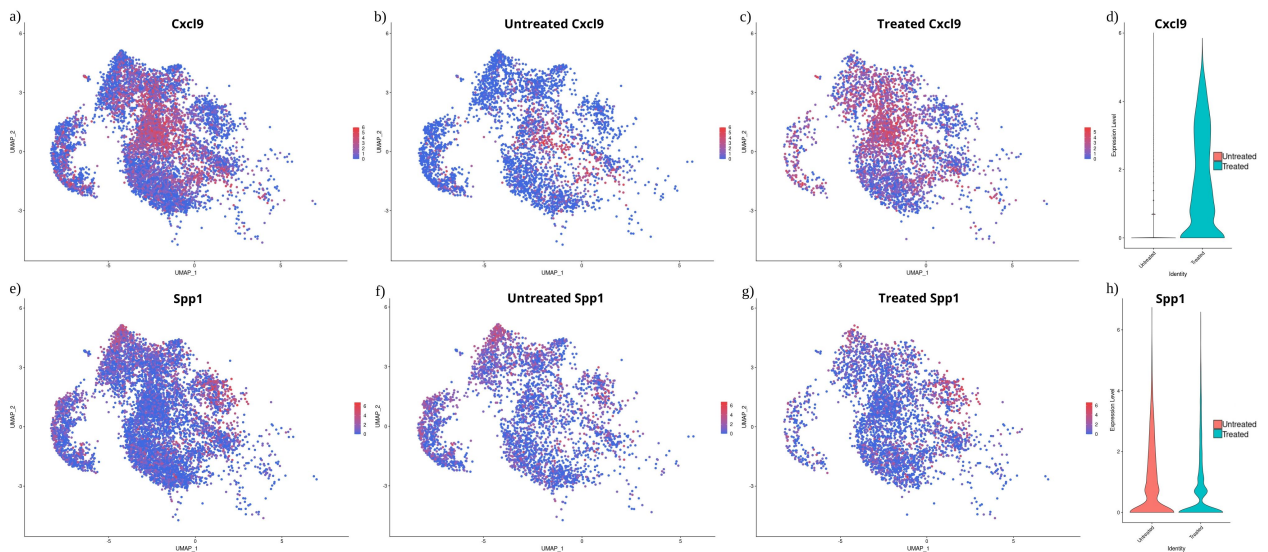


Figure 4.28: Cxcl9 and Spp1 expression analysis in macrophages. a) Cxcl9 Expression in macrophages. b) Cxcl9 expression in Untreated macrophages. c) Cxcl9 expression in Treated macrophages. d) Cxcl9 violin plots divided by Condition. e) Spp1 Expression in macrophages. f) Spp1 expression in Untreated macrophages. g) Spp1 expression in Treated macrophages. h) Spp1 violin plots divided by Condition.

As a chemokine, Cxcl10 is responsible for attracting T cells to the site of inflammation or infection [22]. The expression of this gene is also increased under treatment conditions, as shown in Figure 4.29. Therefore, we can assume that a relevant treatment effect might be the promotion of a shift in the TAMs phenotype towards a more anti-tumoral phenotype, leading to an increase in the recruitment of T cells to the tumor microenvironment, just as cDCs did in the previous analysis.

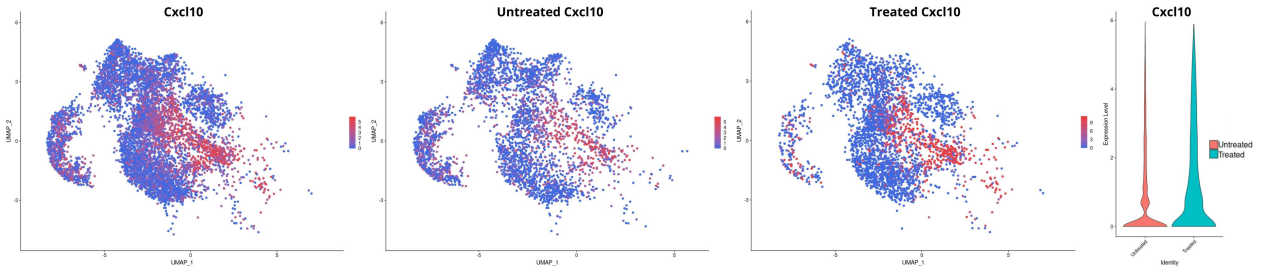


Figure 4.29: Cxcl10 expression analysis in macrophages. a) Cxcl10 Expression in macrophages. b) Cxcl10 expression in Untreated macrophages. c) Cxcl10 expression in Treated macrophages. d) Cxcl10 violin plots divided by Condition

Another point of interest in the macrophage subsets is the Mreg cluster. As mentioned earlier, this group of cells is known to have an immunosuppressive function and tends to exhibit anti-tumoral properties. This is supported by the high expression of the *Spp1* gene even under treatment conditions. However, a relevant qualitative change was also observed, as this cluster showed the highest increase in the expression of the iNOS (*Hmox1*) gene under treatment conditions. This is significant because this gene is often used to classify TAMs as M1-like or antitumoral. Considering this, the most relevant pathways for this cluster were studied, to understand what functions this cluster was fulfilling in each condition (see Figure 4.30).

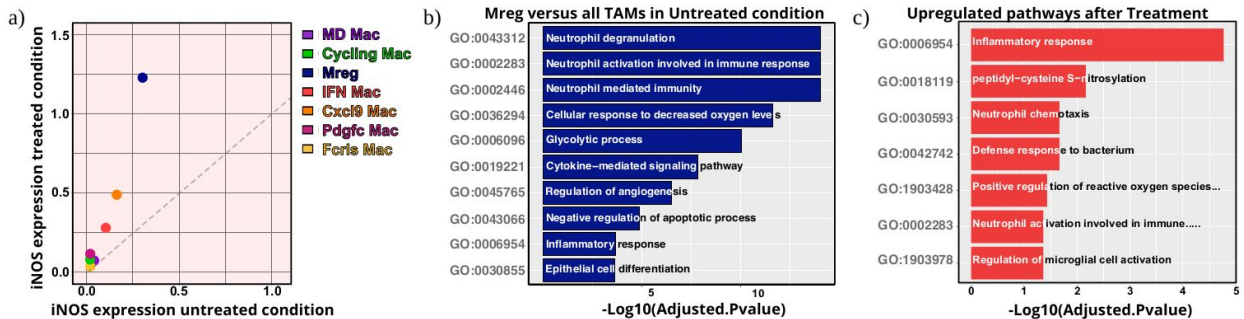


Figure 4.30: Mreg subpopulation analysis. a) Comparison of iNOS expression in each macrophage cluster under treated and untreated conditions. b) Upregulated pathways in Mreg cluster under untreated conditions. c) Upregulated pathways in Mreg cluster after treatment condition in comparison with the untreated condition.

In both cases, Mregs seem to be related to neutrophil recruitment or activation, which could then lead to immunosuppression by neutrophil differentiation into MDSCs [35]. This shows that, even under the effects of the therapy this specific cluster doesn't change its pro-tumoral phenotype and that iNOS expression isn't necessarily a marker of pro-tumoral phenotype.

4.3. Granulocytes

For this analysis, no further quality control was performed for granulocytes. However, a granulocytic cluster was identified between the monocyte/macrophage data. This cluster was separated from the rest of the data and merged with the previous granulocyte dataset. Then, just like in the previous analysis, the merged dataset was integrated by sample (see Figure 4.31). The code utilized for this analysis with the detailed methods utilized can be found in Appendix E.

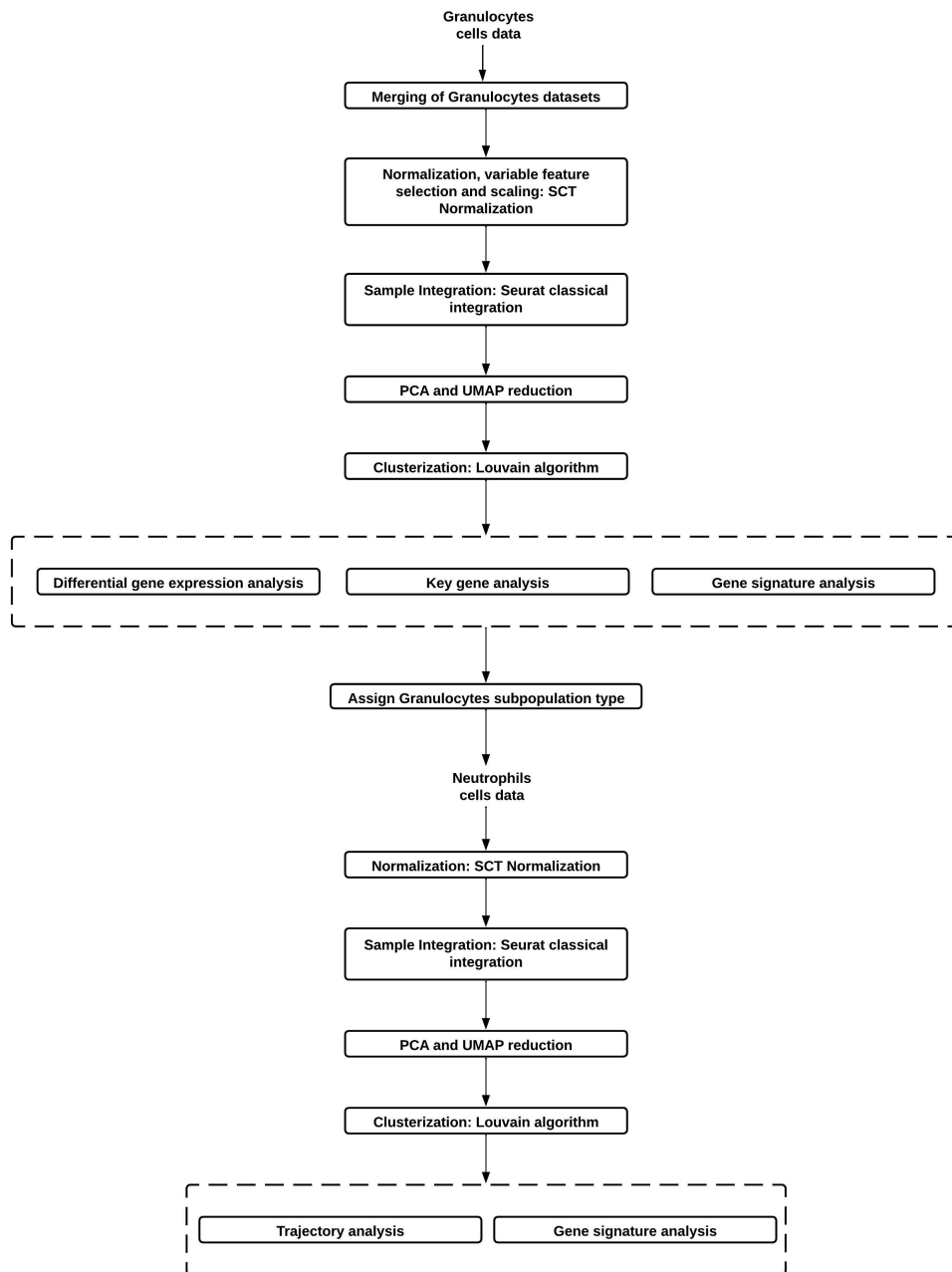


Figure 4.31: Granulocytes pipeline

For PCA and UMAP reductions 40 PC's were used (see Figure 4.21).

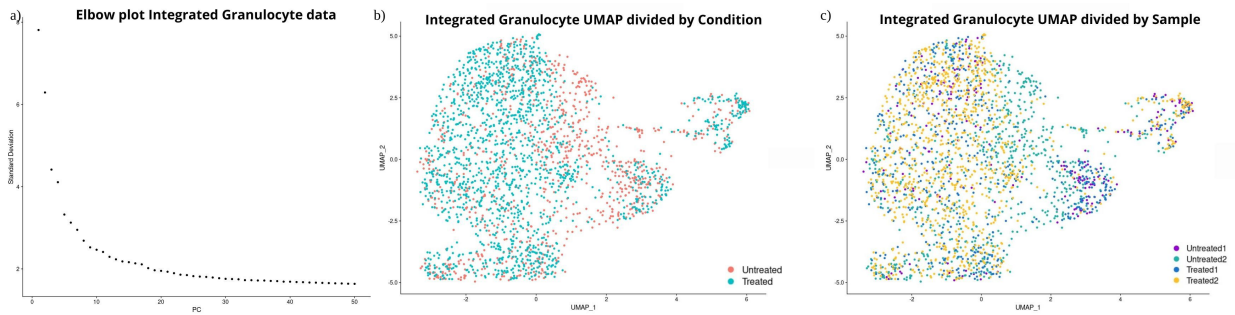


Figure 4.32: Granulocytes data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples.

Analyzing the resulting clustree (see Figure 4.33), it can be seen that there is a clear separation between three populations that do not mix with each other, starting even from resolution 0.1.

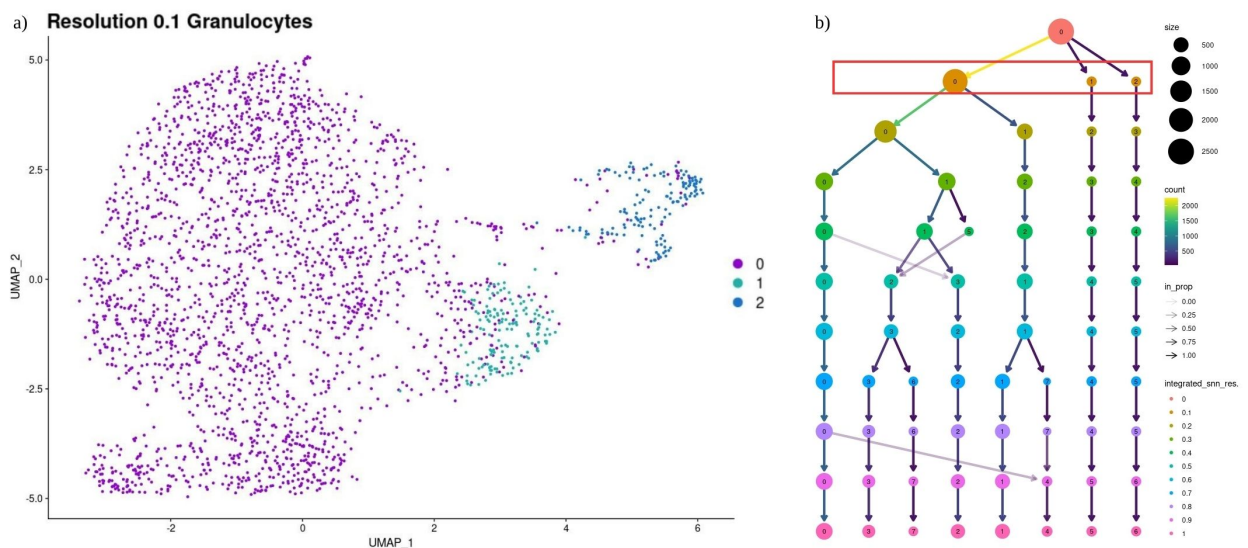


Figure 4.33: Granulocytes clustering. a) Granulocytes UMAP resolution 0.1. b) Granulocytes Clustree.

Considering the above, for the identification of the different subpopulations, the expression of a Neutrophil signature was studied [21], as this is one of the most common granulocyte populations. According to this result, cluster number 0 seemed to be the principal neutrophil cluster, however, cluster 2 also had a mild expression of the Neutrophil signature (see Figure 4.34).

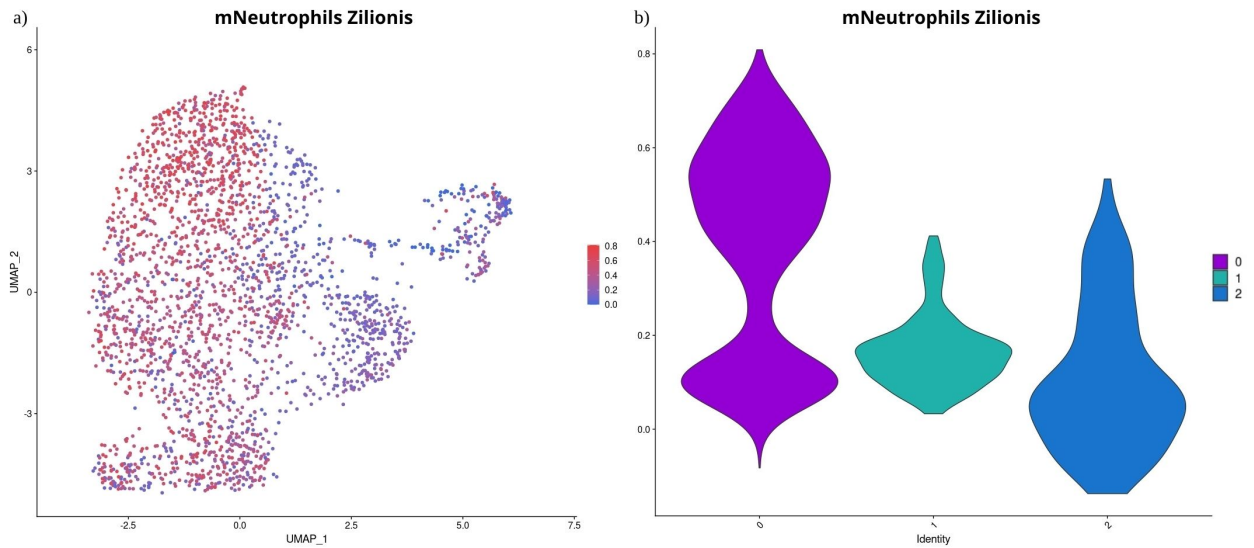


Figure 4.34: Zilionis mNeutrophils signature analysis. a) mNeutrophils signature expression level. b) mNeutrophils signature violin plot

To better understand each cluster, a differential gene expression analysis was conducted (see Figure 4.35). Cluster 2 is very interesting as its most relevant genes correspond to ribosomal genes, something quite particular.

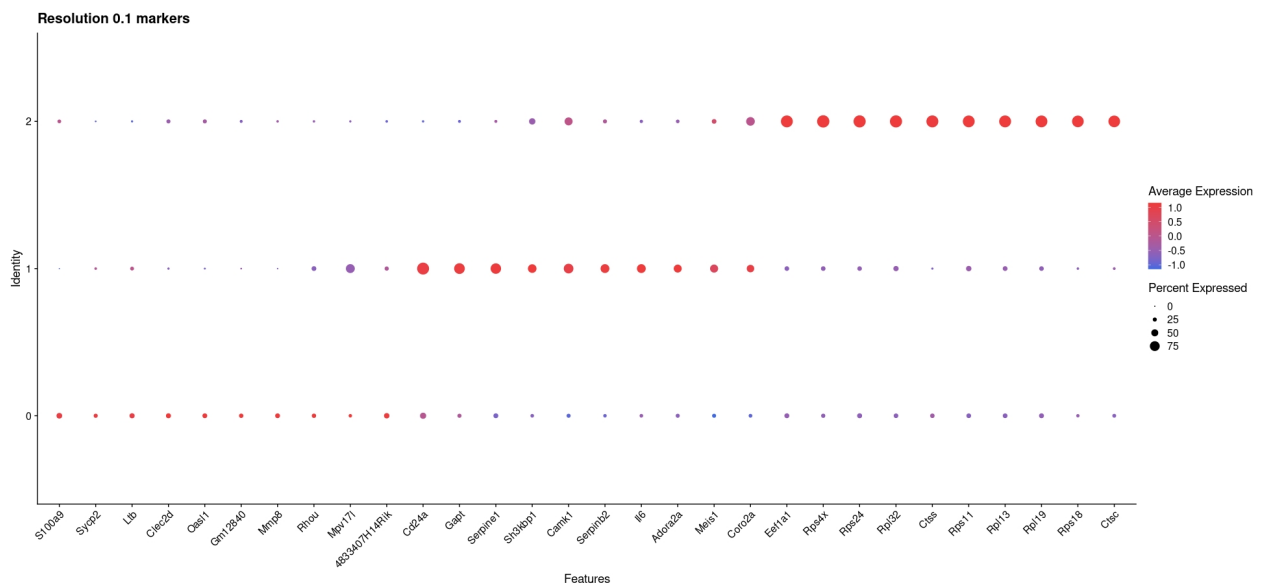


Figure 4.35: Granulocytes resolution 0.1 Dotplot

Through more bibliographical research, it was discovered that the observed characteristics of cluster 2 could be aligned with a specific type of neutrophils known as Transcriptionally active neutrophils [36]. The signature for this type of neutrophils was analyzed and the results indicated that cluster 2 indeed exhibited a high expression for this particular signature (see

Figure 4.36). Therefore, the cluster was classified as such.

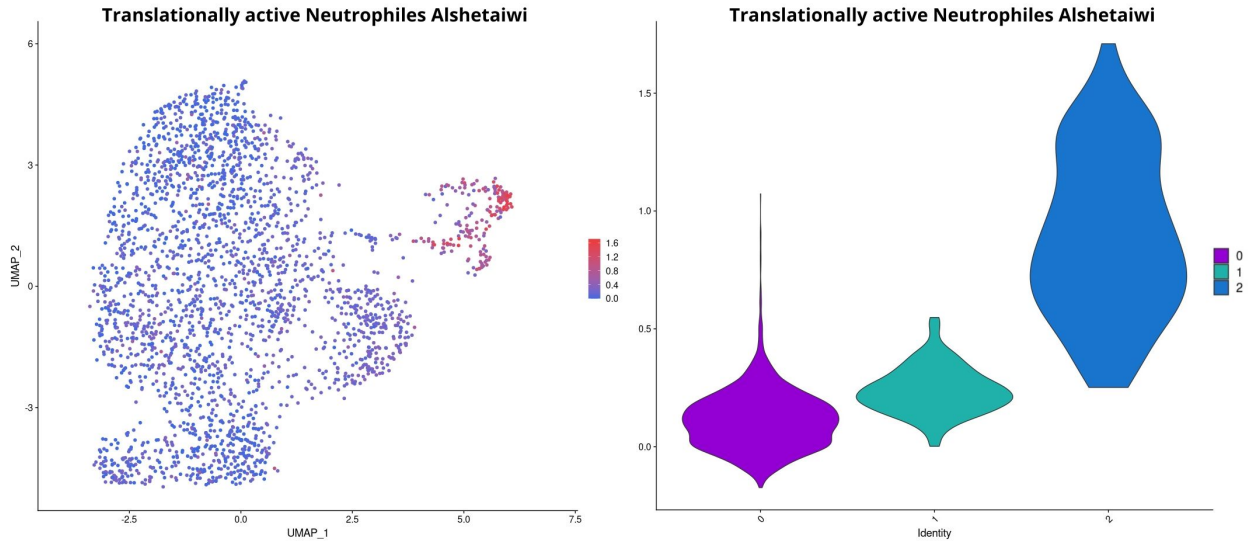


Figure 4.36: Alshetaiwi Transcriptionally active neutrophils signature analysis. a) Transcriptionally active neutrophils signature expression level. b) Transcriptionally active neutrophils signature violin plot

For the final cluster, cluster number 1, an eosinophil signature, composed by known eosinophil markers *Siglec f*, *Il5ra*, *Ccr3* and *Epx* [57]. Cluster 1 had indeed a high expression for this signature 4.37, therefore this cluster was classified as eosinophils. It is important to note that a particular sector of Cluster 0, which corresponds to neutrophils, also showed a high expression for the eosinophil signature. Upon further investigation, it was discovered that certain subsets of neutrophils exhibit eosinophil markers, and some eosinophils can also express Neutrophil markers [58]. Considering the divisions observed in the clustree, it was determined that this group of cells were most likely neutrophils expressing eosinophil markers, since cluster 1 stays as an isolated cluster from other cells, without dividing itself or mixing with other clusters, while cluster 0 in higher resolutions divides itself into different subsets. Therefore it is more probable that these cells correspond to a subset of neutrophils rather than eosinophils.

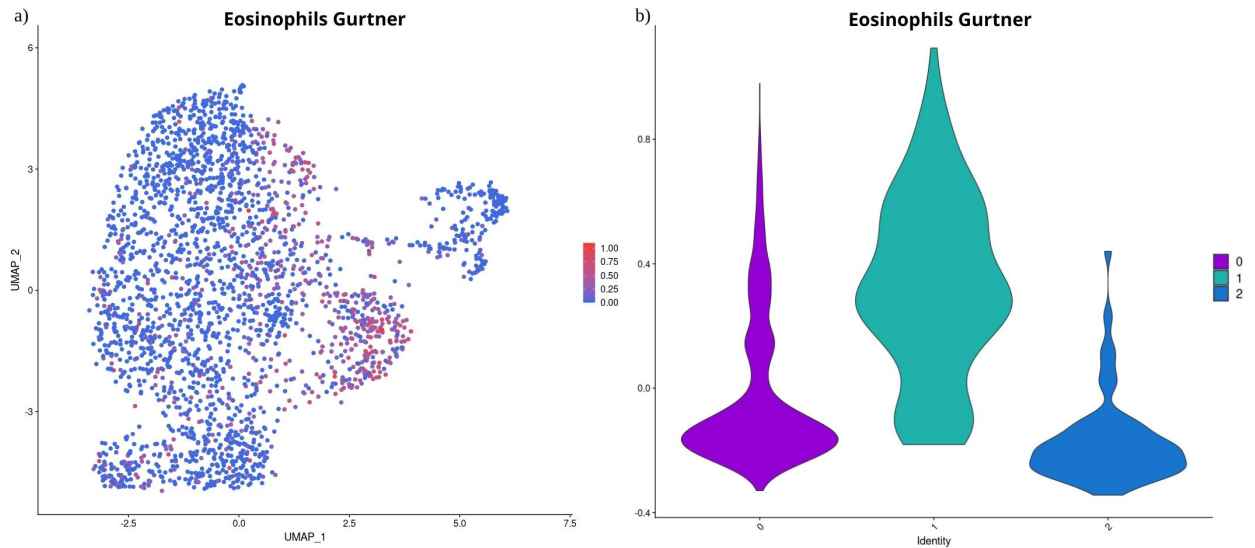


Figure 4.37: Gurtner eosinophils key genes signature analysis. a) Eosinophils key genes expression level. b) Eosinophils key genes violin plot

With this final analysis all subpopulations were identified now (see Figure 4.38). In the matter of quantitative changes, generally, all three populations increased under the treatment condition (see Figure 4.39), with Translationally active neutrophils being the highest, with a 1.25 log₂FC (see Appendix F.4).

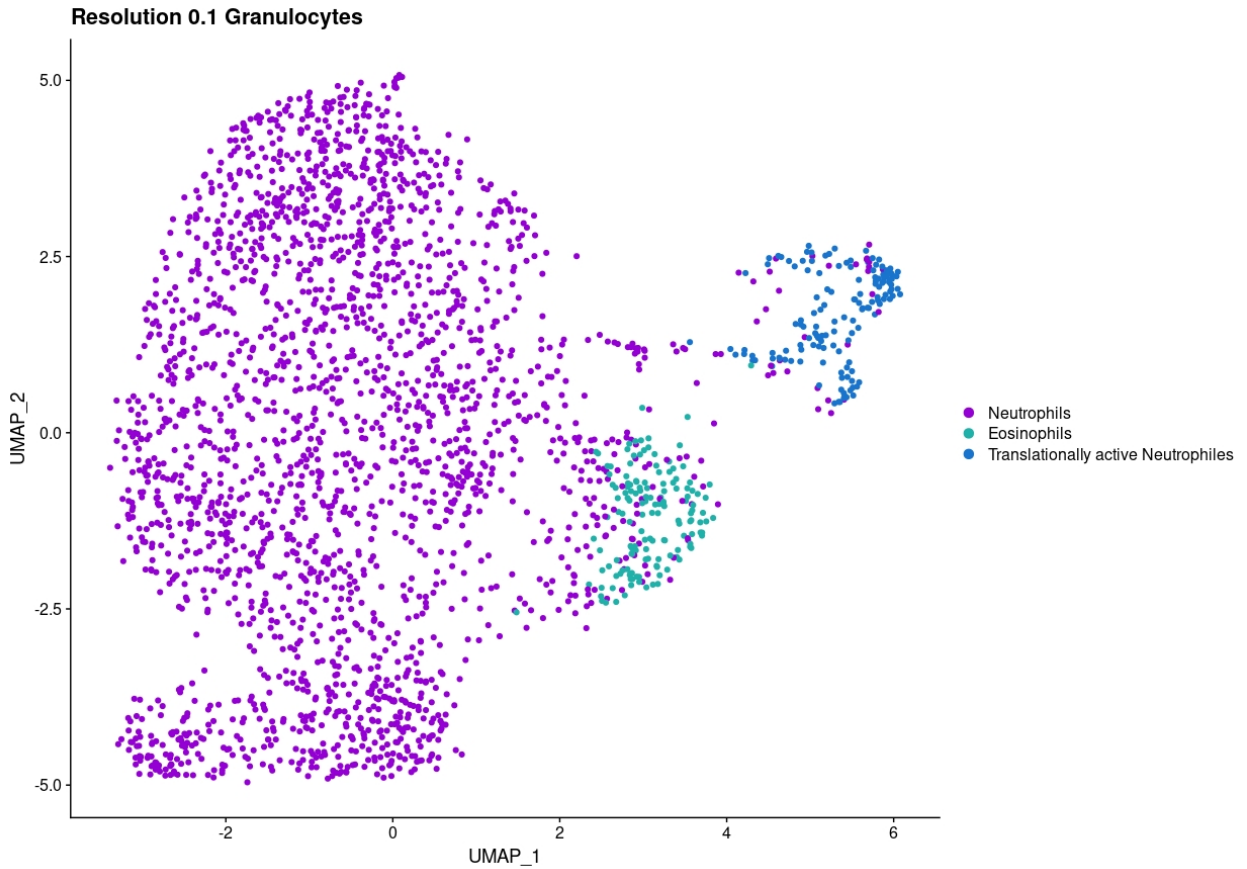


Figure 4.38: Granulocytes subpopulations

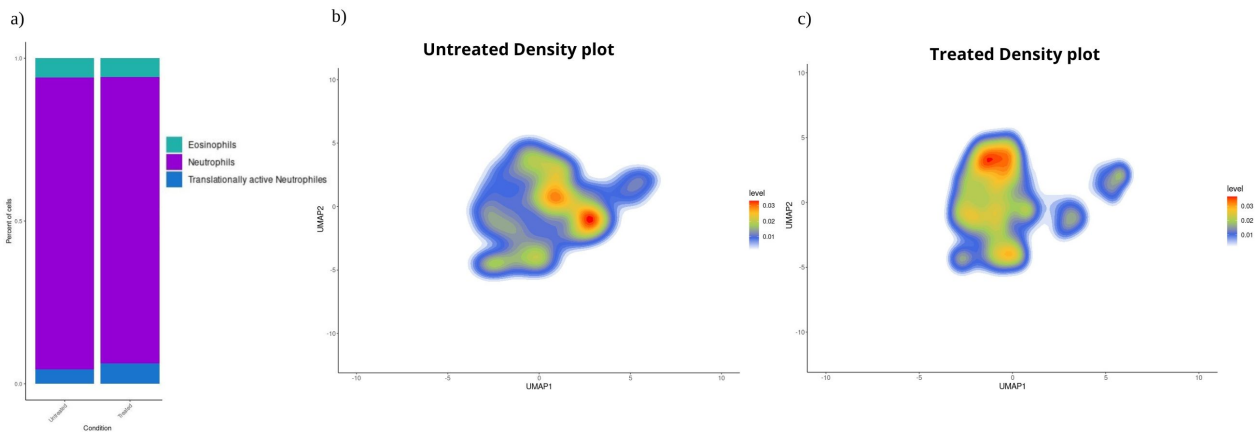


Figure 4.39: Granulocytes subpopulations changes. a) Barplot of granulocytes subpopulations percent by Conditions. b) Untreated granulocytes density plots. c) Treated granulocytes density plots

More important than the changes at this level, was to analyze what was happening with the neutrophils in term of maturation and if MDSCs, more specifically granulocytic MDSCs (G-MDSCs), were present in this group. In order to do this, the Neutrophil cluster was

separated from the other cells and re-integrated. For this re-integration, some modifications were necessary due to the low quantity of cells and the tendency for neutrophils to express fewer genes. Specifically, the analysis methods had to be adjusted by reducing the number of genes used for normalization was reduced to 2000 and the number of PCs used was 30 (see Figure 4.40)

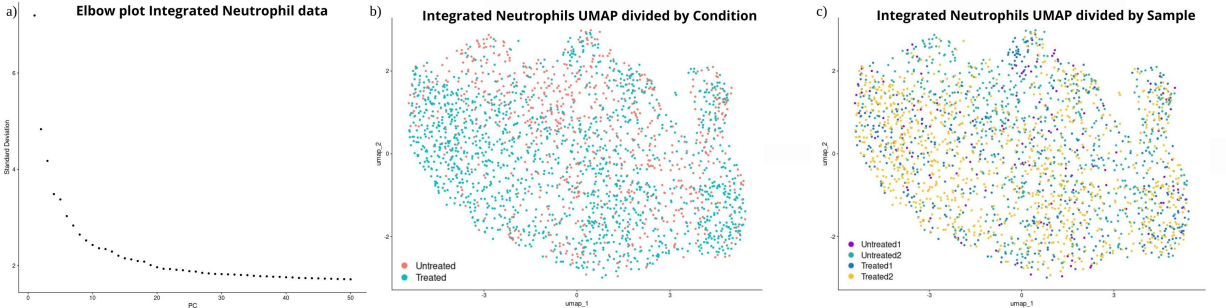


Figure 4.40: Neutrophils data graphics. a) Elbow plot. b) UMAP divided by Condition. c) UMAP divided by Samples.

One advantage of working with a lower number of cells was that it was possible to construct a Monocle2 trajectory. For this, first, it was necessary to determine a resolution stable enough to use as the basis for the trajectory. In this case, the chosen resolution was 0.7 (see Figure 4.41)

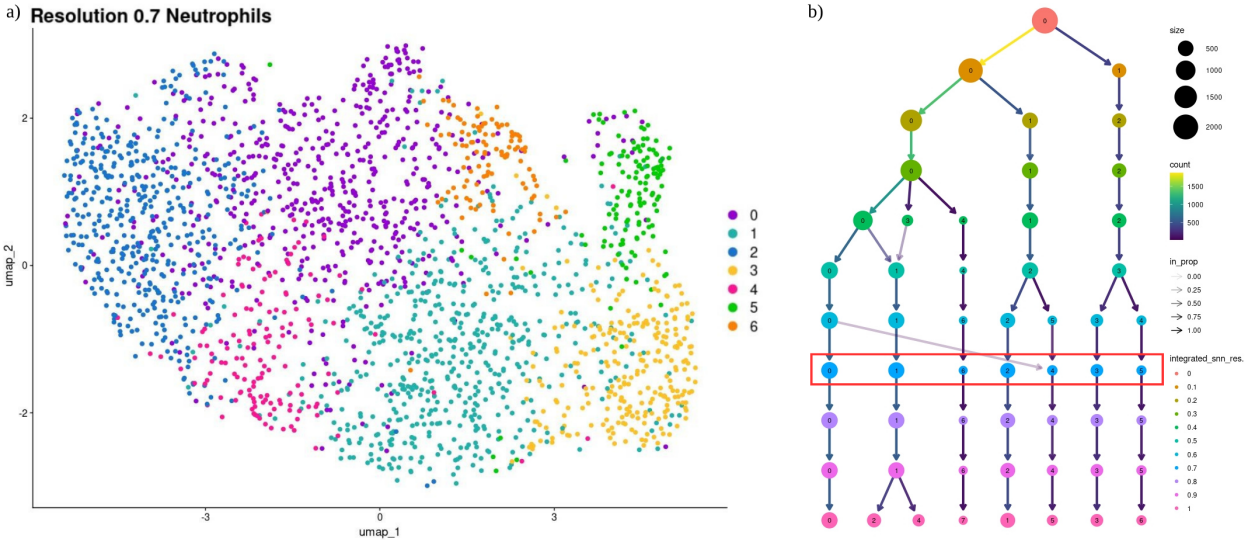


Figure 4.41: Neutrophils clustering. a) Neutrophils UMAP resolution 0.1. b) Neutrophils Clustree.

The resulting trajectory shows a group of cells in an initial state, referred to as State 1 by Monocle, which then branches into two different states (refer to Figure 4.42). This indicates

that there is not only one maturation path of neutrophils, but also an alternative maturation path that could correspond to G-MDSCs. The cells in State 2, suspected G-MDSCs, have a lower Pseudotime than State 3, indicating that they are a less mature group of cells. This aligns with what has been observed between G-MDSCs and neutrophils, with the latter being the more mature of the two [36].

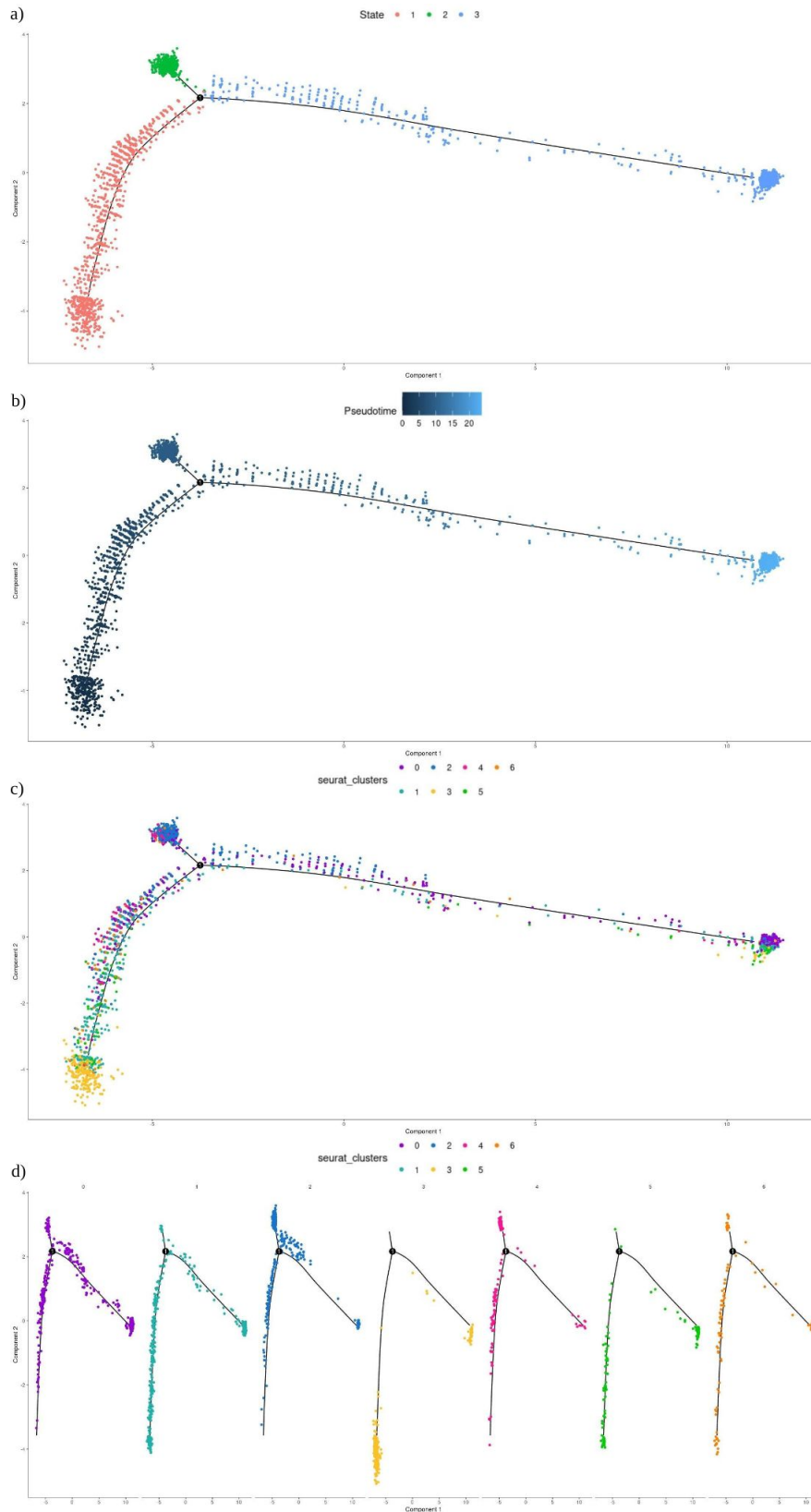


Figure 4.42: Neutrophil trajectory using Monocle 2. a) Neutrophils trajectory colored by State. b) Neutrophils trajectory colored by Pseudotime. c) Neutrophils trajectory colored by Seurat Clusters. d) Neutrophils trajectory divided by Seurat Clusters.

Using a mice signature for G-MDSCs [36], it was possible to verify that indeed State corresponded to this population, as it was the only one with a high expression of it. In Seurat's UMAP, this population corresponds to the lower left side of the group (see Figure 4.43).

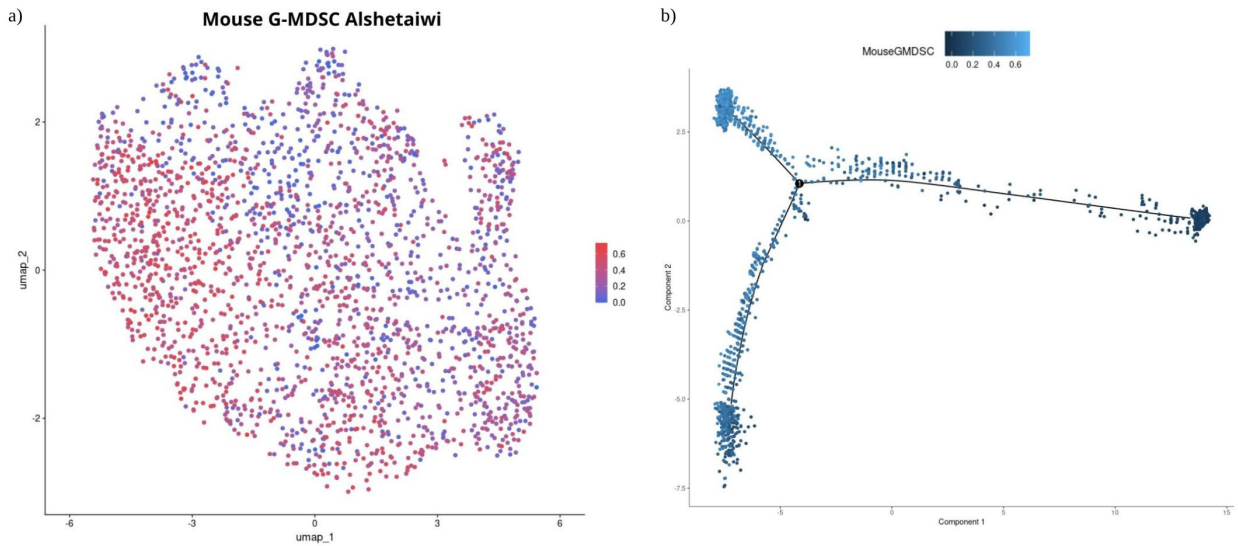


Figure 4.43: Alshetaiwi mouse G-MDSCs signature analysis. a) Mouse G-MDSCs expression level. b) Mouse G-MDSCs expression level on Monocle2 trajectory graphic

When translating the information obtained through Monocle2 to Seurat, it was decided that, instead of using clusters to identify the different subpopulations, Monocle States would be used instead. This decision was made because the most distinguishing characteristic of this group of cells corresponds to their different maturation states. Furthermore, when viewing the Monocle2 data in UMAP, the division done by Monocle was understandable in this graphic too, with the three States clearly located in different positions inside the UMAP, alongside the distribution of the Pseudotime (see Figure 4.44).

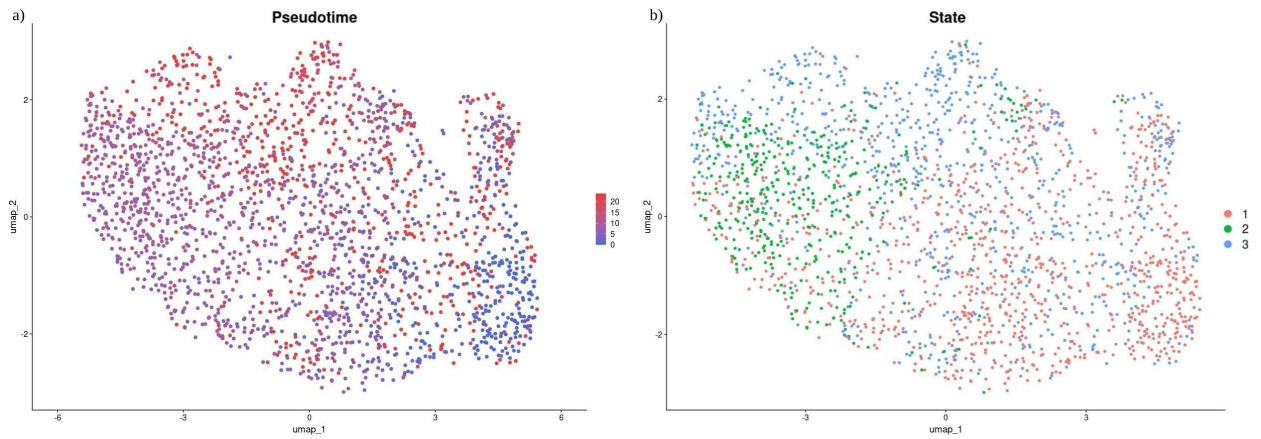


Figure 4.44: Monocle2 data on UMAP. a) Neutrophils Monocle2 States. b) Neutrophils Monocle2 Pseudotime

With the division decided, the only thing left was to classify the different States. Based on the analysis done, State 2 would correspond to G-MDSCs, while 1 and 3 to neutrophils states with different levels of maturation. Considering this, a Progenitor neutrophil signature was analyzed, however, no specific State or section of the UMAP presented a high enough level of expression for a Progenitor neutrophil signature to be considered as such (see Figure 4.45). This does make sense since none of the samples analyzed corresponded to cells from bone marrow, where progenitor cells like Progenitor neutrophils tend to be found [36].

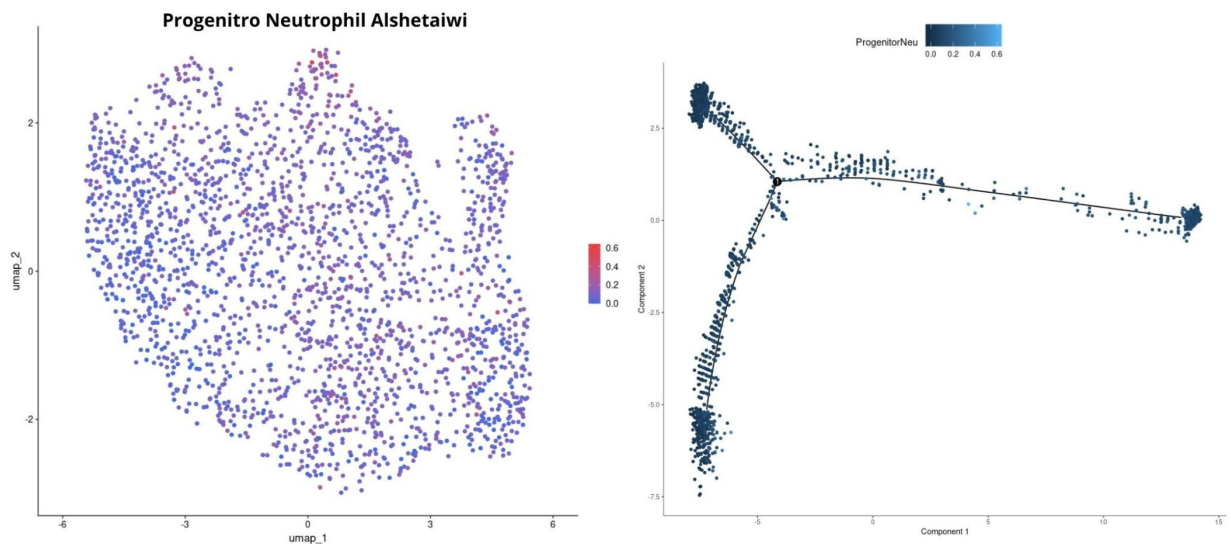


Figure 4.45: Alshetaiwi Progenitor neutrophil signature analysis. a) Progenitor neutrophil expression level. b) Progenitor neutrophil expression level on Monocle2 trajectory graphic

Considering the above, it was decided to classify States 1 and 3 based on their respective

maturation levels. Since State 3 was determined to be more mature than State 1, it was classified as Mature neutrophils. State 1, on the other hand, was only classified as neutrophils, indicating that they are not fully mature, but not progenitor cells either (see Figure 4.46).

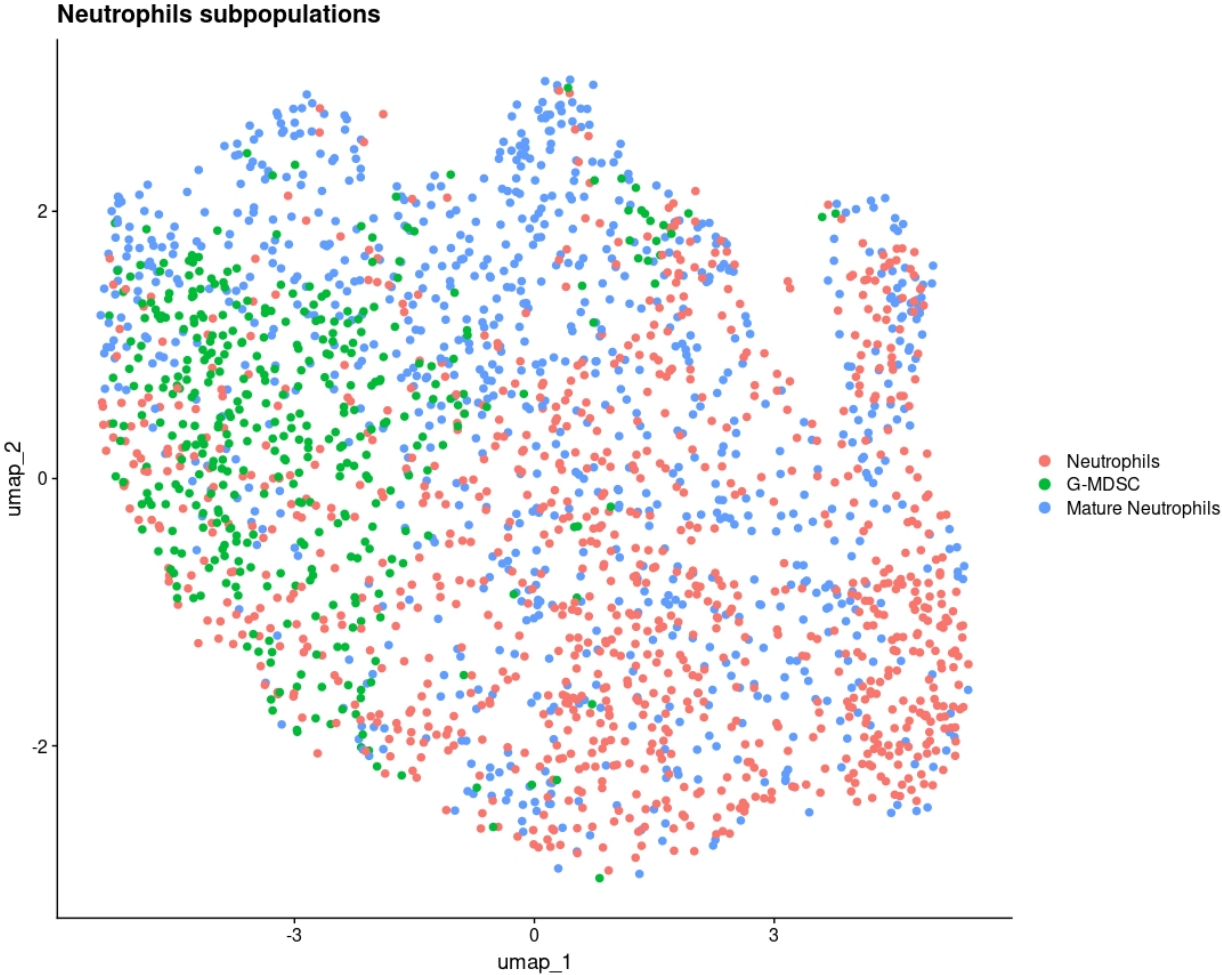


Figure 4.46: Neutrophils subpopulations

The observed quantitative changes between conditions are very drastic, with both neutrophils and G-MDSCs increasing their proportion by around 5 log₂FC in treated condition, while Mature neutrophils had a decrease of -2.97 in their number (see Appendix F.5). Based on the obtained trajectory, it appears that in untreated or normal conditions, neutrophils tend to mature more into the Mature neutrophil state and only a few of them differentiate into G-MDSCs. However, when treated, a significant number of cells remain at the neutrophil state and those that do mature further tend to differentiate into G-MDSCs instead of Mature neutrophils (see Figure 4.47).

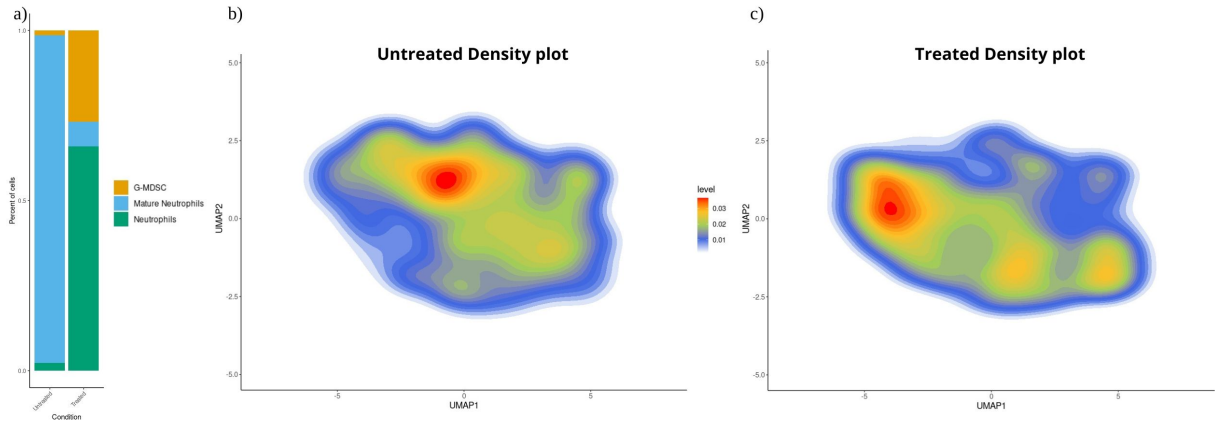


Figure 4.47: Neutrophils subpopulations changes. a) Barplot of neutrophils subpopulations percent by Conditions. b) Untreated neutrophils density plots. c) Treated neutrophils density plots

In terms of qualitative changes, an increase in the population of G-MDSCs results in a rise in the immunosuppressive phenotype of TME. Based on the findings of previous sections, the increase in both G-MDSCs and DC3 contributes to the immunosuppression of other immune cells, particularly T cells. Regarding the significance of this in terms of response to treatment, these changes may be a response of the immune system to the increased recruitment of T cells to the tumor microenvironment by cDCs and Cxcl9 TAMs. Under normal circumstances, this would maintain a balance between the interacting immune cells. However, in cases such as Rhabdoid tumors, this immunosuppression leads to the exhaustion of T cells, making them ineffective in fighting tumor cells. Therefore, as mentioned in previous sections, for the treatment to function correctly, it should be coupled with treatments aimed at preventing the exhaustion of T cells, such as anti-PD1 [5].

Considering that most changes are related to T cell recruitment, previous results of Institut Curie studies were consulted. It was observed that, under treatment conditions, the population of CD8+ T cells indeed increased. Based on these findings, the results of an analysis, previously conducted in collaboration with Institut Curie, regarding changes in CD8+ cell subpopulations were examined. This analysis showed that a specific type of CD8+ subpopulation, known as progenitor-exhausted T cells, increased with the treatment [59]. This observation is very relevant as this subpopulation is identified as one of the primary targets for anti-PD1 treatment, reinforcing the hypothesis that a combined treatment approach with anti-PD1 can yield favourable outcomes. Both the T cell increase results and the CD8+ subset analysis can be found in appendix G.

5

Conclusions

In the present study the effects of a novel targeted immunotherapy on tumor-infiltrating immune populations were analysed using single-cell RNA sequencing. This specific therapy was selected for the analysis since in previous studies done by Institut Curie it was observed that it was capable of decreasing tumor growth. For the present study, appropriate methods for the analysis of the scRNA-seq data were determined, this was done by designing different pipelines to use for the myeloid populations encountered, with Seurat's pipeline being used as a basis. Details on the pipelines were adjusted accordingly to the type of cells studied, including the number of genes used for normalization, the number of principal components (PCs) and strategies used to identify each subpopulation.

Firstly, the present myeloid populations were classified using key genes *Csf1r*, *Flt3*, and *Mmp9* for monocytes/macrophages, dendritic cells, and granulocytes, respectively. The changes under treated conditions were evaluated, and it was determined that only the granulocyte population increased, while both dendritic cells and monocytes/macrophages decreased.

Next, subpopulation changes were analyzed for each population, identifying them using as a basis the previously known subpopulations. In the case of dendritic cells due to the consistency in dendritic cell subpopulation classification, gene signatures were used to identify each of them. The most relevant changes in the subpopulations were an increase in the immunoregulatory DC3 subpopulation and a decrease in the cDCs subpopulations, which was attributed to the migration of these cells to the lymph nodes for T cell recruitment.

For monocytes/macrophages, key gene expression analysis and a trajectory analysis were done to classify the subpopulations. Additionally, via a differential gene expression analysis a specific macrophage subpopulation with increased expression of *Cxcl9* gene was identified and it was determined that this subpopulation increased under treated conditions. Further analyzing the macrophage subsets, it was observed that under treatment conditions, TAMs, excluding the Mreg subset, had an increased anti-tumoral phenotype, distinguished by a

higher expression of chemokines that would participate in the recruitment of T cells.

Finally, for the granulocyte population, three distinctive subpopulations were identified: neutrophils, Translationally active neutrophils and Eosinophils. The neutrophil population was then further analyzed using Monocle2 for a cell trajectory analysis, in which three different states were recognized. It was observed that neutrophils were taking two different pathways of maturation, which were classified as Mature neutrophils and G-MDSCs, while the less mature state was named neutrophils. Under treatment conditions, the alternative G-MDSC pathway seemed to be preferred over the Mature neutrophil pathway, suggesting an increase in immunosuppressive phenotype in the tumor microenvironment.

Regarding the correlation between changes in the immune populations and the effects of the therapy, given the increased expression of the Cxcl9 gene in macrophages and the decrease in cDC1 and cDC2 dendritic cell subpopulations, it is highly likely that the treatment leads to increased recruitment of T cells to the tumor microenvironment, this being its most important effect. However, with the increase in both DC3 and G-MDSC populations, there is also an increase in immunosuppressive functions, which leads to the exhaustion of T cells and consequently loss of function. Therefore, for better responses to the treatment, it is recommended to apply it in conjunction with other treatments that prevent T cell exhaustion, such as anti-PD1.

6

Future Developments

This research has several opportunities for future developments and further exploration. One way to better understand the underlying effects of this immunotherapy is to characterize the main metabolic pathways that define each of the identified subpopulations in detail. This approach can provide a more complete and precise understanding of the quantitative changes in the subpopulations

Moreover, there is potential to expand and refine the analysis of the granulocyte population done in this work, especially in the case of neutrophils. While the research mainly focused on the G-MDSC population, studying other subpopulations, such as the ones of neutrophils and eosinophils, could reveal additional aspects that are relevant to the overall understanding of the effects of this therapy and the possible responses to the treatment.

To further validate the clinical applicability of these findings, it is suggested to extend the research to human cells. This additional step would allow determining whether the effects observed in the present study also manifest in humans.

Additionally, given the promising results obtained so far, it is highly recommended to conduct a study on the possible impacts of combined treatment with the current therapy and anti-PD1 therapy.

Bibliography

- [1] K. Nemes, P. D. Johann, S. Tüchert, P. Melchior, C. Vokuhl, R. Siebert, R. Furtwängler, and M. C. Frühwald, “Current and Emerging Therapeutic Approaches for Extracranial Malignant Rhabdoid Tumors,” *Cancer Management and Research*, vol. 14, p. 479, 2022. [Online]. Available: [/pmc/articles/PMC8841298/](#)[/pmc/articles/PMC8841298/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC8841298/](#)
- [2] K. Ginn and A. Gajjar, “Atypical Teratoid Rhabdoid Tumor: Current Therapy and Future Directions,” *Frontiers in Oncology*, vol. 2, 2012. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fonc.2012.00114>
- [3] A. Leruste, C. Chauvin, C. Pouponnot, F. Bourdeaut, J. J. Waterfall, and E. Piaggio, “Immune responses in genomically simple SWI/SNF-deficient cancers,” *Cancer*, vol. 127, no. 2, pp. 172–180, 1 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/cncr.33172https://onlinelibrary.wiley.com/doi/abs/10.1002/cncr.33172https://acsjournals.onlinelibrary.wiley.com/doi/10.1002/cncr.33172>
- [4] P. D. Johann, S. Erkek, M. Zapatka, K. Kerl, I. Buchhalter, V. Hovestadt, D. T. W. Jones, D. Sturm, C. Hermann, M. Segura Wang, A. Korshunov, M. Rhyzova, S. Gröbner, S. Brabetz, L. Chavez, S. Bens, S. Gröschel, F. Kratochwil, A. Wittmann, L. Sieber, C. Geörg, S. Wolf, K. Beck, F. Oyen, D. Capper, P. van Sluis, R. Volckmann, J. Koster, R. Versteeg, A. von Deimling, T. Milde, O. Witt, A. E. Kulozik, M. Ebinger, T. Shalaby, M. Grotzer, D. Sumerauer, J. Zamecnik, J. Mora, N. Jabado, M. D. Taylor, A. Huang, E. Aronica, A. Bertoni, B. Radlwimmer, T. Pietsch, U. Schüller, R. Schneppenheim, P. A. Northcott, J. O. Korbel, R. Siebert, M. C. Frühwald, P. Lichter, R. Eils, A. Gajjar, M. Hasselblatt, S. M. Pfister, and M. Kool, “Atypical Teratoid/Rhabdoid Tumors Are Comprised of Three Epigenetic Subgroups with Distinct Enhancer Landscapes,” *Cancer Cell*, vol. 29, no. 3, pp. 379–393, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1535610816300356>
- [5] A. Leruste, J. Tosello, R. N. Ramos, A. Tauziède-Espariat, S. Brohard, Z.-Y. Han, K. Beccaria, M. Andrianteranagna, P. Caudana, J. Nikolic, C. Chauvin, L. L. Niborski, V. Manriquez, W. Richer, J. Masliah-Planchon, S. Grossetête-Lalami, M. Bohec, S. Lameiras, S. Baulande, C. Pouponnot, A. Coulomb, L. Galmiche, D. Surdez, N. Servant, J. Helft, C. Sedlik, S. Puget, P. Benaroch, O. Delattre, J. J. Waterfall, E. Piaggio, and F. Bourdeaut, “Clonally Expanded T Cells Reveal Immunogenicity

- of Rhabdoid Tumors,” *Cancer Cell*, vol. 36, no. 6, pp. 597–612, 12 2019. [Online]. Available: [https://www.cell.com/cancer-cell/fulltext/S1535-6108\(19\)30482-9](https://www.cell.com/cancer-cell/fulltext/S1535-6108(19)30482-9)
- [6] N. Mashtalir, A. R. D’Avino, B. C. Michel, J. Luo, J. Pan, J. E. Otto, H. J. Zullo, Z. M. McKenzie, R. L. Kubiak, R. St. Pierre, A. M. Valencia, S. J. Poynter, S. H. Cassel, J. A. Ranish, and C. Kadoch, “Modular Organization and Assembly of SWI/SNF Family Chromatin Remodeling Complexes,” *Cell*, vol. 175, no. 5, pp. 1272–1288, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0092867418312443>
- [7] R. M. Samstein, C. H. Lee, A. N. Shoushtari, M. D. Hellmann, R. Shen, Y. Y. Janjigian, D. A. Barron, A. Zehir, E. J. Jordan, A. Omuro, T. J. Kaley, S. M. Kendall, R. J. Motzer, A. A. Hakimi, M. H. Voss, P. Russo, J. Rosenberg, G. Iyer, B. H. Bochner, D. F. Bajorin, H. A. Al-Ahmadie, J. E. Chaft, C. M. Rudin, G. J. Riely, S. Baxi, A. L. Ho, R. J. Wong, D. G. Pfister, J. D. Wolchok, C. A. Barker, P. H. Gutin, C. W. Brennan, V. Tabar, I. K. Mellinghoff, L. M. DeAngelis, C. E. Ariyan, N. Lee, W. D. Tap, M. M. Gounder, S. P. D’Angelo, L. Saltz, Z. K. Stadler, H. I. Scher, J. Baselga, P. Razavi, C. A. Klebanoff, R. Yaeger, N. H. Segal, G. Y. Ku, R. P. DeMatteo, M. Ladanyi, N. A. Rizvi, M. F. Berger, N. Riaz, D. B. Solit, T. A. Chan, and L. G. Morris, “Tumor mutational load predicts survival after immunotherapy across multiple cancer types,” *Nature Genetics* 2019 51:2, vol. 51, no. 2, pp. 202–206, 1 2019. [Online]. Available: <https://www.nature.com/articles/s41588-018-0312-8>
- [8] H. J. E. Chun, P. D. Johann, K. Milne, M. Zapatka, A. Buellesbach, N. Ishaque, M. Iskar, S. Erkek, L. Wei, B. Tessier-Cloutier, J. Lever, E. Titmuss, J. T. Topham, R. Bowlby, E. Chuah, K. L. Mungall, Y. Ma, A. J. Mungall, R. A. Moore, M. D. Taylor, D. S. Gerhard, S. J. Jones, A. Korshunov, M. Gessler, K. Kerl, M. Hasselblatt, M. C. Frühwald, E. J. Perlman, B. H. Nelson, S. M. Pfister, M. A. Marra, and M. Kool, “Identification and Analyses of Extra-Cranial and Cranial Rhabdoid Tumor Molecular Subgroups Reveal Tumors with Cytotoxic T Cell Infiltration,” *Cell Reports*, vol. 29, no. 8, pp. 2338–2354, 11 2019. [Online]. Available: <http://www.cell.com/article/S2211124719313130/fulltext><http://www.cell.com/article/S2211124719313130/abstract>[https://www.cell.com/cell-reports/abstract/S2211-1247\(19\)31313-0](https://www.cell.com/cell-reports/abstract/S2211-1247(19)31313-0)
- [9] Z. Y. Han, W. Richer, P. Fréneaux, C. Chauvin, C. Lucchesi, D. Guillemot, C. Grison, D. Lequin, G. Pierron, J. Masliah-Planchon, A. Nicolas, D. Ranchère-Vince, P. Varlet, S. Puget, I. Janoueix-Lerosey, O. Ayrault, D. Surdez, O. Delattre, and F. Bourdeaut, “The occurrence of intracranial rhabdoid tumours in mice depends on temporal control of Smarcb1 inactivation,” *Nature Communications* 2016 7:1, vol. 7, no. 1, pp. 1–11, 1 2016. [Online]. Available: <https://www.nature.com/articles/ncomms10421>
- [10] M. Chaib, S. C. Chauhan, and L. Makowski, “Friend or Foe? Recent Strategies to Target Myeloid Cells in Cancer,” *Frontiers in Cell and Developmental Biology*, vol. 8, 5 2020. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fcell.2020.00351/full>

- [11] C. Engblom, C. Pfirschke, and M. J. Pittet, “The role of myeloid cells in cancer therapies,” *Nature Reviews Cancer* 2016 16:7, vol. 16, no. 7, pp. 447–462, 6 2016. [Online]. Available: <https://www.nature.com/articles/nrc.2016.54>
- [12] S. Cheng, Z. Li, R. Gao, B. Xing, Y. Gao, Y. Yang, S. Qin, L. Zhang, H. Ouyang, P. Du, L. Jiang, B. Zhang, Y. Yang, X. Wang, X. Ren, J.-X. Bei, X. Hu, Z. Bu, J. Ji, and Z. Zhang, “A pan-cancer single-cell transcriptional atlas of tumor infiltrating myeloid cells,” *Cell*, vol. 184, no. 3, pp. 792–809, 2 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0092867421000106>
- [13] F. Ginhoux and S. Jung, “Monocytes and macrophages: developmental pathways and tissue homeostasis,” *Nature Reviews Immunology* 2014 14:6, vol. 14, no. 6, pp. 392–404, 5 2014. [Online]. Available: <https://www.nature.com/articles/nri3671>
- [14] T. S. Kapellos, L. Bonaguro, I. Gemünd, N. Reusch, A. Saglam, E. R. Hinkley, and J. L. Schultze, “Human Monocyte Subsets and Phenotypes in Major Chronic Inflammatory Diseases,” *Frontiers in Immunology*, vol. 10, no. AUG, p. 2035, 2019. [Online]. Available: [/pmc/articles/PMC6728754/](https://pmc/articles/PMC6728754/)[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6728754/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6728754/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6728754/)
- [15] L. M. Carlin, E. G. Stamatiades, C. Auffray, R. N. Hanna, L. Glover, G. Vizcay-Barrena, C. C. Hedrick, H. T. Cook, S. Diebold, and F. Geissmann, “Nr4a1-dependent Ly6C^{low} monocytes monitor endothelial cells and orchestrate their disposal,” *Cell*, vol. 153, no. 2, pp. 362–375, 4 2013. [Online]. Available: [http://www.cell.com/article/S0092867413003322/fulltexthttp://www.cell.com/article/S0092867413003322/abstracthttps://www.cell.com/cell/abstract/S0092-8674\(13\)00332-2](http://www.cell.com/article/S0092867413003322/fulltexthttp://www.cell.com/article/S0092867413003322/abstracthttps://www.cell.com/cell/abstract/S0092-8674(13)00332-2)
- [16] C. Jakubzick, E. L. Gautier, S. L. Gibbings, D. K. Sojka, A. Schlitzer, T. E. Johnson, S. Ivanov, Q. Duan, S. Bala, T. Condon, N. vanRooijen, J. R. Grainger, Y. Belkaid, A. Ma’ayan, D. W. Riches, W. M. Yokoyama, F. Ginhoux, P. M. Henson, and G. J. Randolph, “Minimal differentiation of classical monocytes as they survey steady-state tissues and transport antigen to lymph nodes,” *Immunity*, vol. 39, no. 3, pp. 599–610, 9 2013. [Online]. Available: [http://www.cell.com/article/S107476131300335X/fulltexthttp://www.cell.com/article/S107476131300335X/abstracthttps://www.cell.com/immunity/abstract/S1074-7613\(13\)00335-X](http://www.cell.com/article/S107476131300335X/fulltexthttp://www.cell.com/article/S107476131300335X/abstracthttps://www.cell.com/immunity/abstract/S1074-7613(13)00335-X)
- [17] S. Guo, X. Chen, C. Guo, and W. Wang, “Tumour-associated macrophages heterogeneity drives resistance to clinical therapy,” *Expert Reviews in Molecular Medicine*, vol. 24, p. e17, 4 2022. [Online]. Available: <https://www.cambridge.org/core/journals/expert-reviews-in-molecular-medicine/article/tumour-associated-macrophages-heterogeneity-drive-s-resistance-to-clinical-therapy/B30AEBFC090C3C2F0DD2D8EB9539A8A6>
- [18] R. Y. Ma, A. Black, and B. Z. Qian, “Macrophage diversity in cancer revisited in the era of single-cell omics,” *Trends in Immunology*, vol. 43, no. 7, pp. 546–563, 7 2022.

- [Online]. Available: [http://www.cell.com/article/S1471490622000941/fulltexthttp://www.cell.com/article/S1471490622000941/abstracthttps://www.cell.com/trends/immunology/abstract/S1471-4906\(22\)00094-1](http://www.cell.com/article/S1471490622000941/fulltexthttp://www.cell.com/article/S1471490622000941/abstracthttps://www.cell.com/trends/immunology/abstract/S1471-4906(22)00094-1)
- [19] C. D. Mills, “M1 and M2 Macrophages: Oracles of Health and Disease,” *Critical reviews in immunology*, vol. 32, no. 6, pp. 463–488, 2012. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/23428224/>
- [20] S. Chen, A. F. Saeed, Q. Liu, Q. Jiang, H. Xu, G. G. Xiao, L. Rao, and Y. Duo, “Macrophages in immunoregulation and therapeutics,” *Signal Transduction and Targeted Therapy* 2023 8:1, vol. 8, no. 1, pp. 1–35, 5 2023. [Online]. Available: <https://www.nature.com/articles/s41392-023-01452-1>
- [21] R. Zilionis, C. Engblom, C. Pfirschke, V. Savova, D. Zemmour, H. D. Saatcioglu, I. Krishnan, G. Maroni, C. V. Meyerovitz, C. M. Kerwin, S. Choi, W. G. Richards, A. De Rienzo, D. G. Tenen, R. Bueno, E. Levantini, M. J. Pittet, and A. M. Klein, “Single-Cell Transcriptomics of Human and Mouse Lung Cancers Reveals Conserved Myeloid Populations across Individuals and Species,” *Immunity*, vol. 50, no. 5, pp. 1317–1334, 5 2019. [Online]. Available: [http://www.cell.com/article/S1074761319301268/fulltexthttp://www.cell.com/article/S1074761319301268/abstracthttps://www.cell.com/immunity/abstract/S1074-7613\(19\)30126-8](http://www.cell.com/article/S1074761319301268/fulltexthttp://www.cell.com/article/S1074761319301268/abstracthttps://www.cell.com/immunity/abstract/S1074-7613(19)30126-8)
- [22] R. Bill, P. Wirapati, M. Messemaker, W. Roh, B. Zitti, F. Duval, M. Kiss, J. C. Park, T. M. Saal, J. Hoelzl, D. Tarussio, F. Benedetti, S. Tissot, L. Kandalaft, M. Varrone, G. Ciriello, T. A. McKee, Y. Monnier, M. Mermoud, E. M. Blaum, I. Gushterova, A. L. Gonye, N. Hacohen, G. Getz, T. R. Mempel, A. M. Klein, R. Weissleder, W. C. Faquin, P. M. Sadow, D. Lin, S. I. Pai, M. Sade-Feldman, and M. J. Pittet, “CXCL9:SPP1 macrophage polarity identifies a network of cellular programs that control human cancers,” *Science*, vol. 381, no. 6657, pp. 515–524, 8 2023. [Online]. Available: <https://www.science.org/doi/10.1126/science.ade2292>
- [23] C. Wu, Y. Xue, P. Wang, L. Lin, Q. Liu, N. Li, J. Xu, and X. Cao, “IFN- γ Primes Macrophage Activation by Increasing Phosphatase and Tensin Homolog via Downregulation of miR-3473b,” *The Journal of Immunology*, vol. 193, no. 6, pp. 3036–3044, 9 2014. [Online]. Available: <https://dx.doi.org/10.4049/jimmunol.1302379>
- [24] Y. Katzenelenbogen, F. Sheban, A. Yalin, I. Yofe, D. Svetlichnyy, D. A. Jaitin, C. Bornstein, A. Moshe, H. Keren-Shaul, M. Cohen, S. Y. Wang, B. Li, E. David, T. M. Salame, A. Weiner, and I. Amit, “Coupled scRNA-Seq and Intracellular Protein Activity Reveal an Immunosuppressive Role of TREM2 in Cancer,” *Cell*, vol. 182, no. 4, pp. 872–885, 8 2020. [Online]. Available: [http://www.cell.com/article/S0092867420308096/fulltexthttp://www.cell.com/article/S0092867420308096/abstracthttps://www.cell.com/cell/abstract/S0092-8674\(20\)30809-6](http://www.cell.com/article/S0092867420308096/fulltexthttp://www.cell.com/article/S0092867420308096/abstracthttps://www.cell.com/cell/abstract/S0092-8674(20)30809-6)

- [25] A. C. Villani, R. Satija, G. Reynolds, S. Sarkizova, K. Shekhar, J. Fletcher, M. Griesbeck, A. Butler, S. Zheng, S. Lazo, L. Jardine, D. Dixon, E. Stephenson, E. Nilsson, I. Grundberg, D. McDonald, A. Filby, W. Li, P. L. De Jager, O. Rozenblatt-Rosen, A. A. Lane, M. Haniffa, A. Regev, and N. Hacohen, “Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors,” *Science*, vol. 356, no. 6335, 4 2017. [Online]. Available: <https://www.science.org/doi/10.1126/science.aah4573>
- [26] C. Bosteels, K. Neyt, M. Vanheerswynghels, M. J. van Helden, D. Sichien, N. Debeuf, S. De Prijck, V. Bosteels, N. Vandamme, L. Martens, Y. Saeys, E. Louagie, M. Lesage, D. L. Williams, S. C. Tang, J. U. Mayer, F. Ronchese, C. L. Scott, H. Hammad, M. Guilliams, and B. N. Lambrecht, “Inflammatory Type 2 cDCs Acquire Features of cDC1s and Macrophages to Orchestrate Immunity to Respiratory Virus Infection,” *Immunity*, vol. 52, no. 6, pp. 1039–1056, 6 2020. [Online]. Available: <http://www.cell.com/article/S1074761320301631/fulltext><http://www.cell.com/article/S1074761320301631/abstract>[https://www.cell.com/immunity/abstract/S1074-7613\(20\)30163-1](https://www.cell.com/immunity/abstract/S1074-7613(20)30163-1)
- [27] B. Maier, A. M. Leader, S. T. Chen, N. Tung, C. Chang, J. LeBerichel, A. Chudnovskiy, S. Maskey, L. Walker, J. P. Finnigan, M. E. Kirkling, B. Reizis, S. Ghosh, N. R. D’Amore, N. Bhardwaj, C. V. Rothlin, A. Wolf, R. Flores, T. Marron, A. H. Rahman, E. Kenigsberg, B. D. Brown, and M. Merad, “A conserved dendritic-cell regulatory program limits antitumour immunity,” *Nature* 2020 580:7802, vol. 580, no. 7802, pp. 257–262, 3 2020. [Online]. Available: <https://www.nature.com/articles/s41586-020-2134-y>
- [28] R. A. Backer, H. C. Probst, and B. E. Clausen, “Classical DC2 subsets and monocyte-derived DC: Delineating the developmental and functional relationship,” *European Journal of Immunology*, vol. 53, no. 3, p. 2149548, 3 2023. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/eji.202149548><https://onlinelibrary.wiley.com/doi/abs/10.1002/eji.202149548><https://onlinelibrary.wiley.com/doi/10.1002/eji.202149548>
- [29] G. M. Gerhard, R. Bill, M. Messemaker, A. M. Klein, and M. J. Pittet, “Tumor-infiltrating dendritic cell states are conserved across solid human cancers,” *Journal of Experimental Medicine*, vol. 218, no. 1, 1 2021. [Online]. Available: <https://doi.org/10.1084/jem.20200264>
- [30] Z. Liu, H. Wang, Z. Li, R. J. Dress, Y. Zhu, S. Zhang, D. De Feo, W. T. Kong, P. Cai, A. Shin, C. Piot, J. Yu, Y. Gu, M. Zhang, C. Gao, L. Chen, H. Wang, M. Vétillard, P. Guermonprez, I. Kwok, L. G. Ng, S. Chakarov, A. Schlitzer, B. Becher, C. A. Dutertre, B. Su, and F. Ginhoux, “Dendritic cell type 3 arises from Ly6C+ monocyte-dendritic cell progenitors,” *Immunity*, vol. 56, no. 8, pp. 1761–1777, 8 2023.

- [31] E. Marmonti, J. Oliva-Ramirez, and C. Haymaker, “Dendritic Cells: The Long and Evolving Road towards Successful Targetability in Cancer,” *Cells* 2022, Vol. 11, Page 3028, vol. 11, no. 19, p. 3028, 9 2022. [Online]. Available: <https://www.mdpi.com/2073-4409/11/19/3028/html><https://www.mdpi.com/2073-4409/11/19/3028>
- [32] A. Del Prete, V. Salvi, A. Soriani, M. Laffranchi, F. Sozio, D. Bosisio, and S. Sozzani, “Dendritic cell subsets in cancer immunity and tumor antigen sensing,” *Cellular & Molecular Immunology* 2023 20:5, vol. 20, no. 5, pp. 432–447, 3 2023. [Online]. Available: <https://www.nature.com/articles/s41423-023-00990-6>
- [33] A. E. Marciscano and N. Anandasabapathy, “The Role of Dendritic Cells in Cancer and Anti-Tumor Immunity,” *Seminars in immunology*, vol. 52, p. 101481, 2 2021. [Online]. Available: [/pmc/articles/PMC8545750/](https://pubmed.ncbi.nlm.nih.gov/3545750/)[/pmc/articles/PMC8545750/?report=abstract](https://pubmed.ncbi.nlm.nih.gov/3545750/?report=abstract)
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8545750/>
- [34] D. F. Quail, B. Amulic, M. Aziz, B. J. Barnes, E. Eruslanov, Z. G. Fridlender, H. S. Goodridge, Z. Granot, A. Hidalgo, A. Huttenlocher, M. J. Kaplan, I. Malanchi, T. Merghoub, E. Meylan, V. Mittal, M. J. Pittet, A. Rubio-Ponce, I. A. Udalova, T. K. van den Berg, D. D. Wagner, P. Wang, A. Zychlinsky, K. E. de Visser, M. Egeblad, and P. Kuberski, “Neutrophil phenotypes and functions in cancer: A consensus statement,” *Journal of Experimental Medicine*, vol. 219, no. 6, 6 2022.
- [35] D. I. Gabrilovich, “Myeloid-derived suppressor cells,” *Cancer immunology research*, vol. 5, no. 1, p. 3, 1 2017. [Online]. Available: [/pmc/articles/PMC5426480/](https://pubmed.ncbi.nlm.nih.gov/35426480/)[/pmc/articles/PMC5426480/?report=abstract](https://pubmed.ncbi.nlm.nih.gov/35426480/?report=abstract)
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5426480/>
- [36] H. Alshetaiwi, N. Pervolarakis, L. L. McIntyre, D. Ma, Q. Nguyen, J. A. Rath, K. Nee, G. Hernandez, K. Evans, L. Torosian, A. Silva, C. Walsh, and K. Kessenbrock, “Defining the emergence of myeloid-derived suppressor cells in breast cancer using single-cell transcriptomics,” *Science immunology*, vol. 5, no. 44, 2 2020. [Online]. Available: [/pmc/articles/PMC7219211/](https://pubmed.ncbi.nlm.nih.gov/37219211/)[/pmc/articles/PMC7219211/?report=abstract](https://pubmed.ncbi.nlm.nih.gov/37219211/?report=abstract)
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7219211/>
- [37] X. Li and C.-Y. Wang, “From bulk, single-cell to spatial RNA sequencing,” *International Journal of Oral Science*, vol. 13, no. 1, p. 36, 2021. [Online]. Available: <https://doi.org/10.1038/s41368-021-00146-0>
- [38] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija, “Comprehensive integration of single-cell data,” *Cell*, vol. 177, no. 7, p. 1888, 6 2019. [Online]. Available: [/pmc/articles/PMC6687398/](https://pubmed.ncbi.nlm.nih.gov/36687398/)[/pmc/articles/PMC6687398/?report=abstract](https://pubmed.ncbi.nlm.nih.gov/36687398/?report=abstract)
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6687398/>

- [39] L. Tian, X. Dong, S. Freytag, K.-A. Lê Cao, S. Su, A. JalalAbadi, D. Amann-Zalcenstein, T. S. Weber, A. Seidi, J. S. Jabbari, S. H. Naik, and M. E. Ritchie, “Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments,” *Nature Methods*, vol. 16, no. 6, pp. 479–487, 6 2019. [Online]. Available: <https://www.nature.com/articles/s41592-019-0425-8>
- [40] P. Hoffman, Satija Lab, and Collaborators, “About Seurat.” [Online]. Available: <https://satijalab.org/seurat/#about-seurat>
- [41] M. B. Cole, D. Risso, A. Wagner, D. DeTomaso, J. Ngai, E. Purdom, S. Dudoit, and N. Yosef, “Performance Assessment and Selection of Normalization Procedures for Single-Cell RNA-Seq,” *Cell Systems*, vol. 8, no. 4, pp. 315–328, 4 2019.
- [42] C. Hafemeister and R. Satija, “Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression,” *Genome Biology*, vol. 20, no. 1, p. 296, 12 2019.
- [43] C. A. Vallejos, J. C. Marioni, and S. Richardson, “BASiCS: Bayesian Analysis of Single-Cell Sequencing Data,” *PLOS Computational Biology*, vol. 11, no. 6, p. e1004333, 6 2015.
- [44] R. Bacher, L.-F. Chu, N. Leng, A. P. Gasch, J. A. Thomson, R. M. Stewart, M. Newton, and C. Kendzioriski, “SCnorm: robust normalization of single-cell RNA-seq data,” *Nature Methods*, vol. 14, no. 6, pp. 584–586, 6 2017.
- [45] A. T. L. Lun, K. Bach, and J. C. Marioni, “Pooling across cells to normalize single-cell RNA sequencing data with many zero counts,” *Genome Biology*, vol. 17, no. 1, p. 75, 12 2016.
- [46] V. Y. Kiselev, T. S. Andrews, and M. Hemberg, “Challenges in unsupervised clustering of single-cell RNA-seq data,” *Nature Reviews Genetics* 2018 20:5, vol. 20, no. 5, pp. 273–282, 1 2019. [Online]. Available: <https://www.nature.com/articles/s41576-018-0088-9>
- [47] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev, “Spatial reconstruction of single-cell gene expression data,” *Nature Biotechnology* 2015 33:5, vol. 33, no. 5, pp. 495–502, 4 2015. [Online]. Available: <https://www.nature.com/articles/nbt.3192>
- [48] M. Büttner, Z. Miao, F. A. Wolf, S. A. Teichmann, and F. J. Theis, “A test metric for assessing single-cell RNA-seq batch correction,” *Nature Methods* 2018 16:1, vol. 16, no. 1, pp. 43–49, 12 2018. [Online]. Available: <https://www.nature.com/articles/s41592-018-0254-1>
- [49] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, “Integrating single-cell transcriptomic data across different conditions, technologies, and species,” *Nature Biotechnology* 2018 36:5, vol. 36, no. 5, pp. 411–420, 4 2018. [Online]. Available: <https://www.nature.com/articles/nbt.4096>

- [50] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni, “Batch effects in single-cell RNA sequencing data are corrected by matching mutual nearest neighbours,” *Nature biotechnology*, vol. 36, no. 5, p. 421, 6 2018. [Online]. Available: [/pmc/articles/PMC6152897//pmc/articles/PMC6152897/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6152897/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6152897/)
- [51] R. Satija and Satija Lab and Collaborators, “Seurat - Guided Clustering Tutorial.” [Online]. Available: https://satijalab.org/seurat/articles/pbmc3k_tutorial
- [52] —, “Introduction to scRNA-seq integration.” [Online]. Available: https://satijalab.org/seurat/articles/integration_introduction
- [53] T. Ilicic, J. K. Kim, A. A. Kolodziejczyk, F. O. Bagger, D. J. McCarthy, J. C. Marioni, and S. A. Teichmann, “Classification of low quality cells from single-cell RNA-seq data,” *Genome Biology*, vol. 17, no. 1, p. 29, 2016. [Online]. Available: <https://doi.org/10.1186/s13059-016-0888-1>
- [54] X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. A. Pliner, and C. Trapnell, “Reversed graph embedding resolves complex single-cell trajectories,” *Nature methods*, vol. 14, no. 10, p. 979, 10 2017. [Online]. Available: [/pmc/articles/PMC5764547//pmc/articles/PMC5764547/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC5764547/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5764547/)
- [55] J. Cao, M. Spielmann, X. Qiu, X. Huang, D. M. Ibrahim, A. J. Hill, F. Zhang, S. Mundlos, L. Christiansen, F. J. Steemers, C. Trapnell, and J. Shendure, “The single-cell transcriptional landscape of mammalian organogenesis,” *Nature* 2019 566:7745, vol. 566, no. 7745, pp. 496–502, 2 2019. [Online]. Available: <https://www.nature.com/articles/s41586-019-0969-x>
- [56] W. Ma, J. Lee, D. Backenroth, Y. J. Zhou, E. Bush, P. Sims, K. Liu, and Y. Shen, “Single cell RNA-Seq reveals pre-cDCs fate determined by transcription factor combinatorial dose,” *BMC Molecular and Cell Biology*, vol. 20, no. 1, 6 2019. [Online]. Available: [/pmc/articles/PMC6599345//pmc/articles/PMC6599345/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6599345/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6599345/)
- [57] A. Gurtner, C. Borrelli, I. Gonzalez-Perez, K. Bach, I. E. Acar, N. G. Núñez, D. Crepaz, K. Handler, V. P. Vu, A. Lafzi, K. Stirm, D. Raju, J. Gschwend, K. Basler, C. Schneider, E. Slack, T. Valenta, B. Becher, P. Krebs, A. E. Moor, and I. C. Arnold, “Active eosinophils regulate host defence and immune responses in colitis,” *Nature* 2022 615:7950, vol. 615, no. 7950, pp. 151–157, 12 2022. [Online]. Available: <https://www.nature.com/articles/s41586-022-05628-7>
- [58] S. Berdnikovs, “The twilight zone: plasticity and mixed ontogeny of neutrophil and eosinophil granulocyte subsets,” *Seminars in Immunopathology*, vol. 43, no. 3, pp.

337–346, 6 2021. [Online]. Available: <https://link.springer.com/article/10.1007/s00281-021-00862-z>

- [59] B. C. Miller, D. R. Sen, R. Al Abosy, K. Bi, Y. V. Virkud, M. W. LaFleur, K. B. Yates, A. Lako, K. Felt, G. S. Naik, M. Manos, E. Gjini, J. R. Kuchroo, J. J. Ishizuka, J. L. Collier, G. K. Griffin, S. Maleri, D. E. Comstock, S. A. Weiss, F. D. Brown, A. Panda, M. D. Zimmer, R. T. Manguso, F. S. Hodi, S. J. Rodig, A. H. Sharpe, and W. N. Haining, “Subsets of exhausted CD8+ T cells differentially mediate tumor control and respond to checkpoint blockade,” *Nature immunology*, vol. 20, no. 3, p. 326, 3 2019. [Online]. Available: [/pmc/articles/PMC6673650//pmc/articles/PMC6673650/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6673650/](https://pmc/articles/PMC6673650//pmc/articles/PMC6673650/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6673650/)

ANNEXES

Annex A. Previous results

Previous studies of mice engrafted Rhabdoid tumors done by Curie Institute, using single-cell RNA-seq and bioinformatics tools, have shown that the most common myeloid population present in the tumor microenvironment corresponds to Monocyte and Macrophages (see Figure A.1).

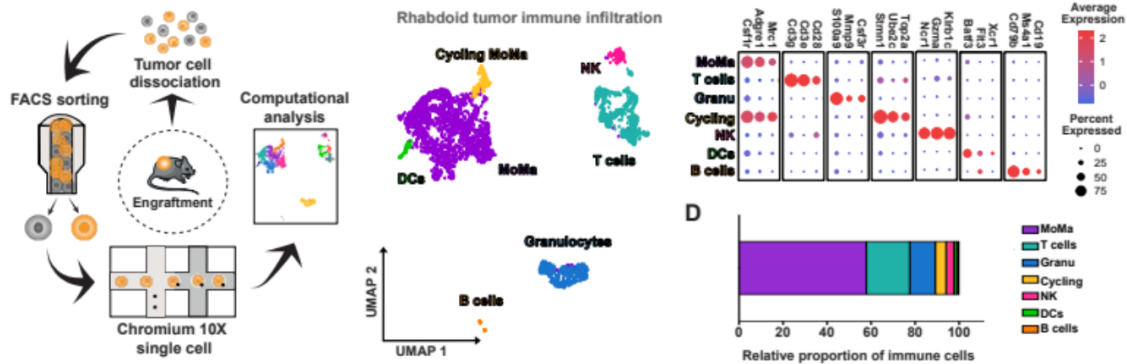


Figure A.1: Analysis of immune populations in the tumor microenvironment. Information provided by Institute Curie

Considering this result, the effects of depleting the Macrophage population were studied (see Figure A.2). The results indicate that depleting the TAMs led to a decrease in tumor growth. However, completely depleting the Macrophage population would pose a significant risk to the patient. Therefore, instead of depleting this population, the focus would be on altering the Macrophages' phenotype to an anti-tumoral one.

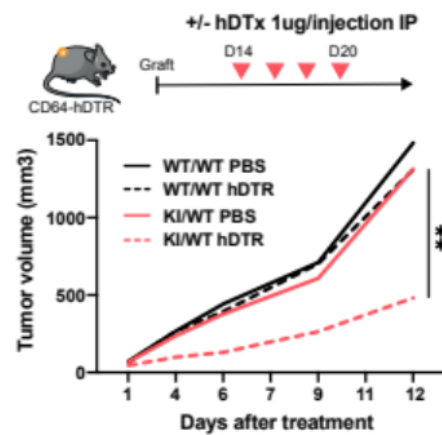


Figure A.2: Analysis of Macrophage depletion in tumor growth. Information provided by Institute Curie

For this, several druggable targets were examined to determine which immune populations could be affected by each target. For the selected target, referred to as “Target x”, it was known that a new immunotherapy affected said target, so a study was conducted to analyse the effect it would have on mice tumor growths (see Figure A.3). The results of the study showed that this treatment helped reduce tumor growth. This immunotherapy corresponds to the therapy analysed in this work, which is why it was chosen for further analysis to determine its effect on the myeloid population.

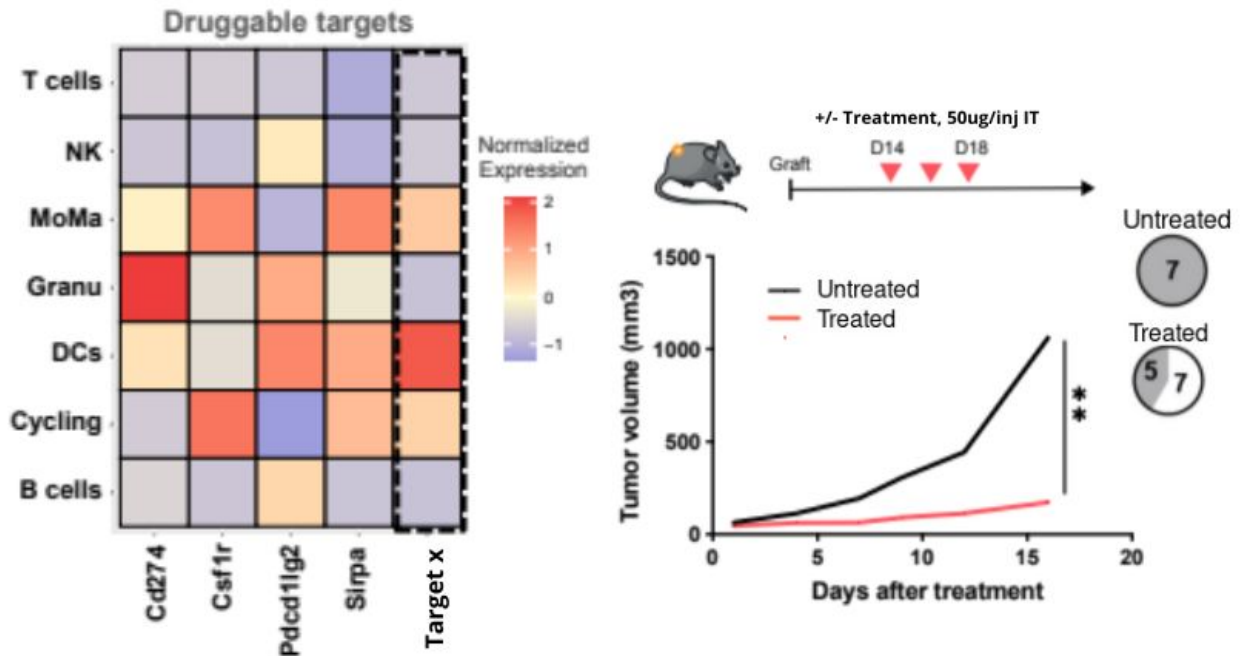


Figure A.3: Analysis of possible druggable targets and study of treatment effect in tumor growth. Information provided by Institute Curie

Annex B. Mice datasets analysis code

Código B.1: R studio code extract – Load required package and establish seed and colors to use

```
1 #---- Preparation ----#
2 #Load required packages
3 library(Seurat)
4 library(clustree)
5 library(dplyr)
6 library(scales)
7 library(ggpubr)
8 library(stringr)
9 library(cowplot)
10 library(ggplot2)
11 library(openxlsx)
12 library(reshape2)
13 library(SeuratWrappers)
14 library(monocle3)
15 library(dittoSeq)
16 library(monocle)
17 library(DDRTree)
18 library(pheatmap)
19
20
21 #set random number generator seed for reproducibility
22 set.seed(123)
23
24 #set colors
25 my_colors=c("darkviolet", "lightseagreen", "dodgerblue3", "↵
↵ goldenrod1", "deeppink", "green3", "darkorange1", "↵
↵ cadetblue4", "mediumvioletred", "brown1", "blue4", "cornsilk4↵
↵ ", "salmon4", "darkred", "aquamarine2", "bisque1")
```

Código B.2: R studio code extract – Load data and create Seurat objects

```
1 ###--- LOAD DATA AND CREATE SEURAT OBJECTS ---###
2
3 #- Load Data -#
4 Untreated1.data <- Read10X(data.dir = "~/Escritorio/Titulo/NGS19↵
↵ -2007/CellRangerCount_Untreated3")
5 Untreated2.data <- Read10X(data.dir = "~/Escritorio/Titulo/NGS19↵
↵ -2007/CellRangerCount_Untreated4")
6 Treated1.data <- Read10X(data.dir = "~/Escritorio/Titulo/NGS19↵
```

```

    ↪ -2007/CellRangerCount_Treated3")
7 Treated2.data <- Read10X(data.dir = "~/Escritorio/Titulo/NGS19↪
    ↪ -2007/CellRangerCount_Treated4")
8
9 #- Create Seurat objects -#
10 Untreated1 <- CreateSeuratObject(counts = Untreated1.data, project↪
    ↪ ="Untreated1", min.cells=0, min.features=200)
11 Untreated2 <- CreateSeuratObject(counts = Untreated2.data, project↪
    ↪ ="Untreated2", min.cells=0, min.features=200)
12 Treated1 <- CreateSeuratObject(counts = Treated1.data, project="↪
    ↪ Treated1", min.cells=0, min.features=200)
13 Treated2 <- CreateSeuratObject(counts = Treated2.data, project="↪
    ↪ Treated2", min.cells=0, min.features=200)

```

Código B.3: R studio code extract – Add Treatment Condition information to the samples

```

1 #- Add "Treatment" info -#
2 Untreated1[["Treatment"]] <- "Untreated"
3 Untreated2[["Treatment"]] <- "Untreated"
4 Treated1[["Treatment"]] <- "Treated"
5 Treated2[["Treatment"]] <- "Treated"
6
7 #- First save -#
8 saveRDS(Untreated1, "Untreated1.rds")
9 saveRDS(Untreated2, "Untreated2.rds")
10 saveRDS(Treated1, "Treated1.rds")
11 saveRDS(Treated2, "Treated2.rds")

```

Código B.4: R studio code extract – Quality Control of Samples

```

1
2 #- Untreated 1 -#
3 #Calculate mitochondrial percent of genes
4 Untreated1[["percent.mt"]] <- PercentageFeatureSet(Untreated1, ↪
    ↪ pattern="^mt-")
5
6 #Logarithmic data
7 logMT <- FetchData(Untreated1, vars=c("percent.mt"))
8 nFeature <- FetchData(Untreated1, vars=c("nFeature_RNA"))
9
10 Untreated1[["logMT"]] <- log10(logMT)
11 Untreated1[["logFeature"]] <- log10(nFeature)

```

```

12
13 #Scatter plots
14 p1 <- FeatureScatter(Untreated1, feature1="nFeature_RNA", feature2=<-
  ↪ ="percent.mt") + geom_hline(yintercept=10) + ggtitle('↪
  ↪ Untreated sample 1')
15 p2 <- FeatureScatter(Untreated1, feature1="logFeatures", feature2=<-
  ↪ "logMT") + geom_hline(yintercept=1) + ggtitle('Untreated ↪
  ↪ sample 1')
16
17 p1+p2
18
19 #Create subset with new data
20 Untreated1 <- subset(Untreated1, subset = logMT < 1)
21
22
23 #- Untreated 2 -#
24 #Calculate mitochondrial percent of genes
25 Untreated1[["percent.mt"]] <- PercentageFeatureSet(Untreated1, ↪
  ↪ pattern="^mt-")
26
27 #Logarithmic data
28 logMT <- FetchData(Untreated1, vars=c("percent.mt"))
29 nFeature <- FetchData(Untreated1, vars=c("nFeature_RNA"))
30
31 Untreated1[["logMT"]] <- log10(logMT)
32 Untreated1[["logFeature"]] <- log10(nFeature)
33
34 #Scatter plots
35 p1 <- FeatureScatter(Untreated2, feature1="nFeature_RNA", feature2=<-
  ↪ ="percent.mt") + geom_hline(yintercept=10) + ggtitle('↪
  ↪ Untreated sample 2')
36 p2 <- FeatureScatter(Untreated2, feature1="logFeatures", feature2=<-
  ↪ "logMT") + geom_hline(yintercept=1) + ggtitle('Untreated ↪
  ↪ sample 2')
37
38 p1+p2
39
40 #Create subset with new data
41 Untreated2 <- subset(Untreated2, subset = logMT < 1)
42
43
44 #- Treated 1 -#
45 #Calculate mitochondrial percent of genes
46 Untreated1[["percent.mt"]] <- PercentageFeatureSet(Untreated1, ↪
  ↪ pattern="^mt-")

```

```

47
48 #Logarithmic data
49 logMT <- FetchData(Untreated1, vars=c("percent.mt"))
50 nFeature <- FetchData(Untreated1, vars=c("nFeature_RNA"))
51
52 Untreated1[["logMT"]] <- log10(logMT)
53 Untreated1[["logFeature"]] <- log10(nFeature)
54
55 #Scatter plots
56 p1 <- FeatureScatter(Treated1, feature1="nFeature_RNA", feature2="↵
↵ percent.mt") + geom_hline(yintercept=7) + ggtitle('Treated ↵
↵ sample 1')
57 p2 <- FeatureScatter(Treated1, feature1="logFeatures", feature2="↵
↵ logMT") + geom_hline(yintercept=0.85) + ggtitle('Treated ↵
↵ sample 1')
58 p1+p2
59
60 #Create subset with new data
61 Treated1 <- subset(Treated1, subset = logMT < 0.85)
62
63
64 #- Treated 2 -#
65 #Calculate mitochondrial percent of genes
66 Untreated1[["percent.mt"]] <- PercentageFeatureSet(Untreated1, ↵
↵ pattern="^mt-")
67
68 #Logarithmic data
69 logMT <- FetchData(Untreated1, vars=c("percent.mt"))
70 nFeature <- FetchData(Untreated1, vars=c("nFeature_RNA"))
71
72 Untreated1[["logMT"]] <- log10(logMT)
73 Untreated1[["logFeature"]] <- log10(nFeature)
74
75 #Scatter plots
76 p1 <- FeatureScatter(Treated2, feature1="nFeature_RNA", feature2="↵
↵ percent.mt") + geom_hline(yintercept=8) + ggtitle('Treated ↵
↵ sample 2')
77 p2 <- FeatureScatter(Treated2, feature1="logFeatures", feature2="↵
↵ logMT") + geom_hline(yintercept=0.9) + ggtitle('Treated ↵
↵ sample 2')
78 p1+p2
79
80 #Create subset with new data
81 Treated2 <- subset(Treated2, subset = logMT < 0.9)
82

```

```

83 #- Save samples -#
84 saveRDS(Untreated1, "Untreated1Flt.rds")
85 saveRDS(Untreated2, "Untreated2Flt.rds")
86 saveRDS(Treated1, "Treated1Flt.rds")
87 saveRDS(Treated2, "Treated2Flt.rds")

```

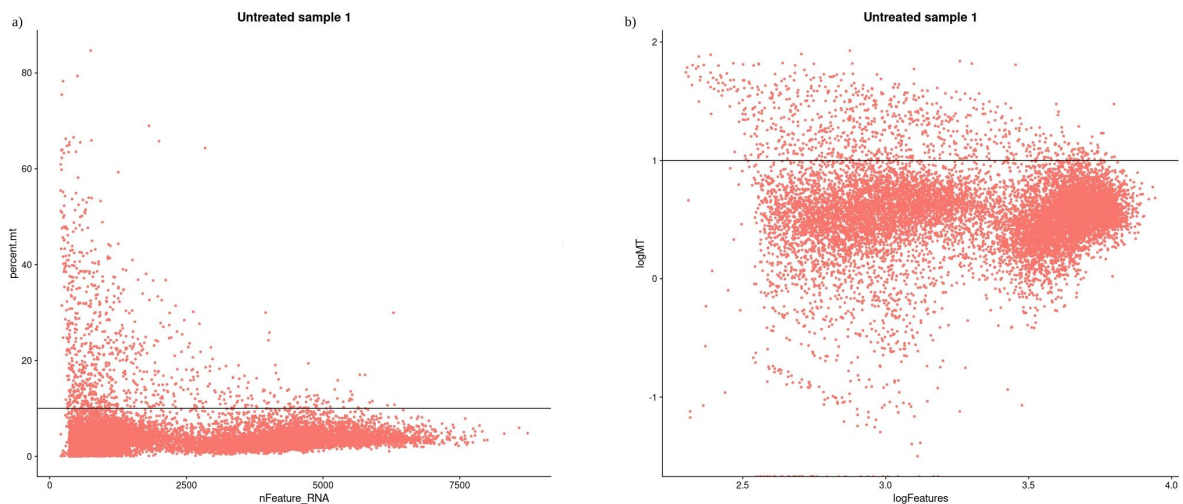


Figure B.1: Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

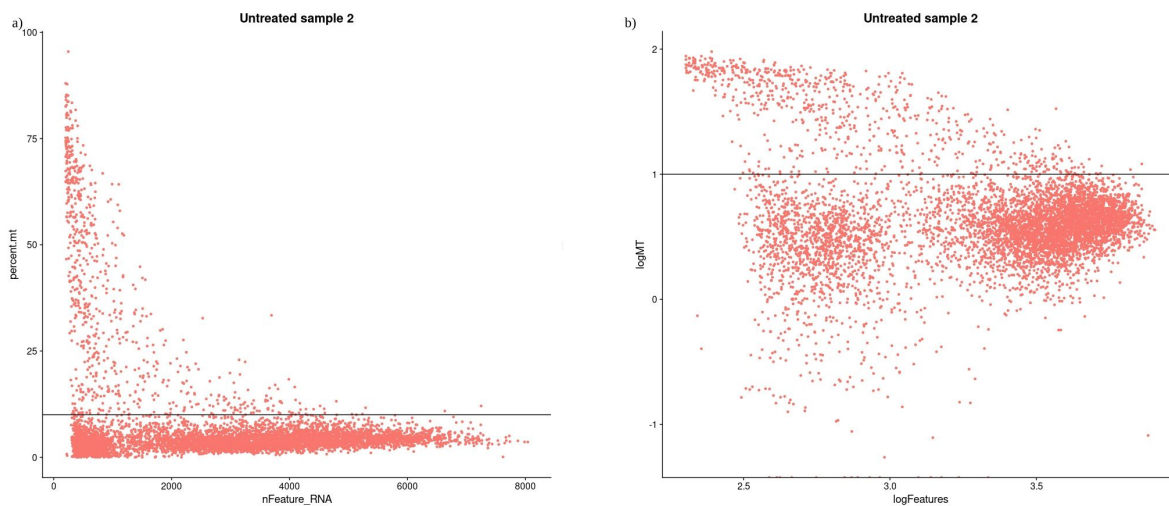


Figure B.2: Scatter plots of mitochondrial gene content versus number of features expressed for Untreated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

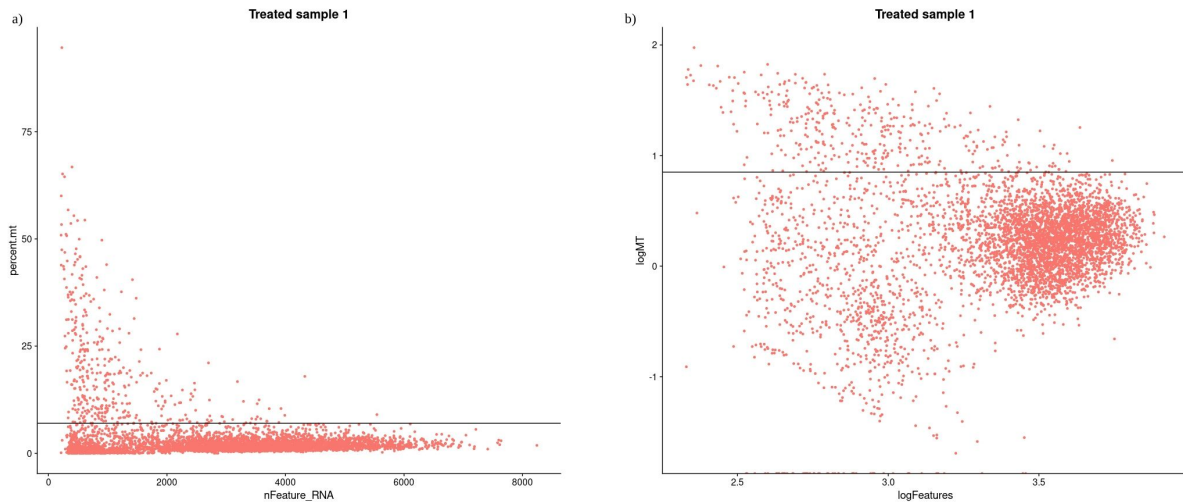


Figure B.3: Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 1. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

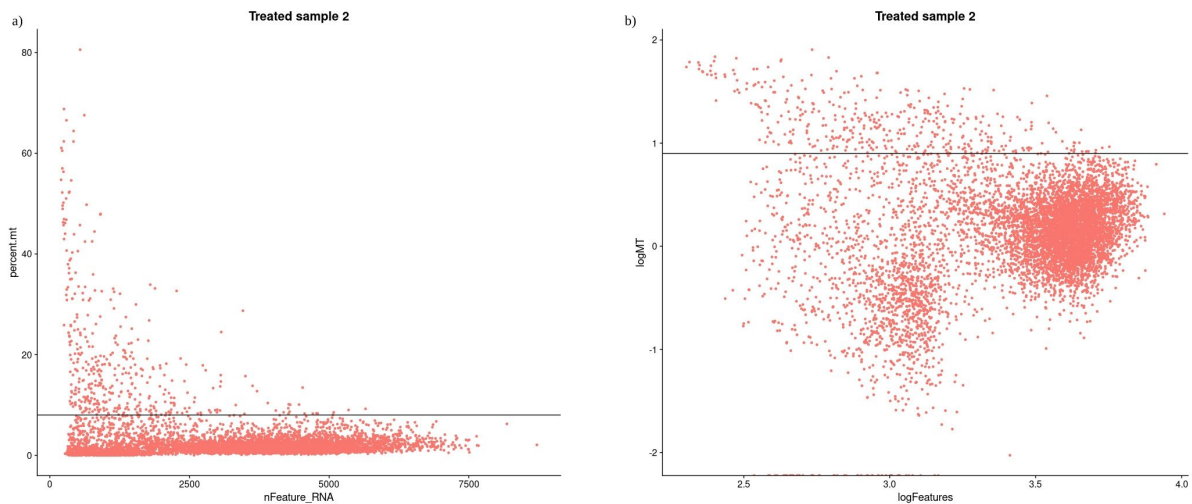


Figure B.4: Scatter plots of mitochondrial gene content versus number of features expressed for Treated sample 2. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

Código B.5: R studio code extract – SCT normalization of the Samples

```

1
2 #--- SCT NORMALIZATION ---#
3
4 Untreated1 <- SCTransform(object = get("Untreated1"), vars.to.regress =
  ↪ regress = "percent.mt", verbose = FALSE, variable.features.n =
  ↪ = 3000)
5 Untreated2 <- SCTransform(object = get("Untreated2"), vars.to.regress =
  ↪ regress = "percent.mt", verbose = FALSE, variable.features.n =
  ↪

```

```

    ↪ = 3000)
6 Treated1 <- SCTransform(object = get("Treated1"), vars.to.regress ↪
    ↪ = "percent.mt", verbose = FALSE, variable.features.n = 3000)
7 Treated2 <- SCTransform(object = get("Treated2"), vars.to.regress ↪
    ↪ = "percent.mt", verbose = FALSE, variable.features.n = 3000)
8
9 #- Save normalized samples -#
10 saveRDS(Untreated1, "Untreated1FltNorm.rds")
11 saveRDS(Untreated2, "Untreated2FltNorm.rds")
12 saveRDS(Treated1, "Treated1FltNorm.rds")
13 saveRDS(Treated2, "Treated2FltNorm.rds")

```

Código B.6: R studio code extract – Merging of Samples

```

1
2 ## --- MERGE --- ##
3 samplesMerge <- merge(Untreated1, y=c(Untreated2, Treated1, ↪
    ↪ Treated2), add.cell.ids=c("Untreated1", y = c("Untreated2", ↪
    ↪ "Treated1", "Treated2")), project="Myeloids")
4 samplesMerge <- SCTransform(object = samplesMerge, vars.to.regress ↪
    ↪ = "percent.mt", verbose = FALSE, variable.features.n = ↪
    ↪ 3000)
5
6 saveRDS(samplesMerge, "samplesMerge.rds")

```

Código B.7: R studio code extract – Data linear reduction

```

1 #- PCA reduction -#
2 samplesMerge <- RunPCA(samplesMerge, npcs = 50, verbose=F)
3
4 #Elbow plot
5 ElbowPlot(samplesMerge, ndims = 50, reduction = "pca") + ggtitle("↪
    ↪ Elbow plot Merged data")

```

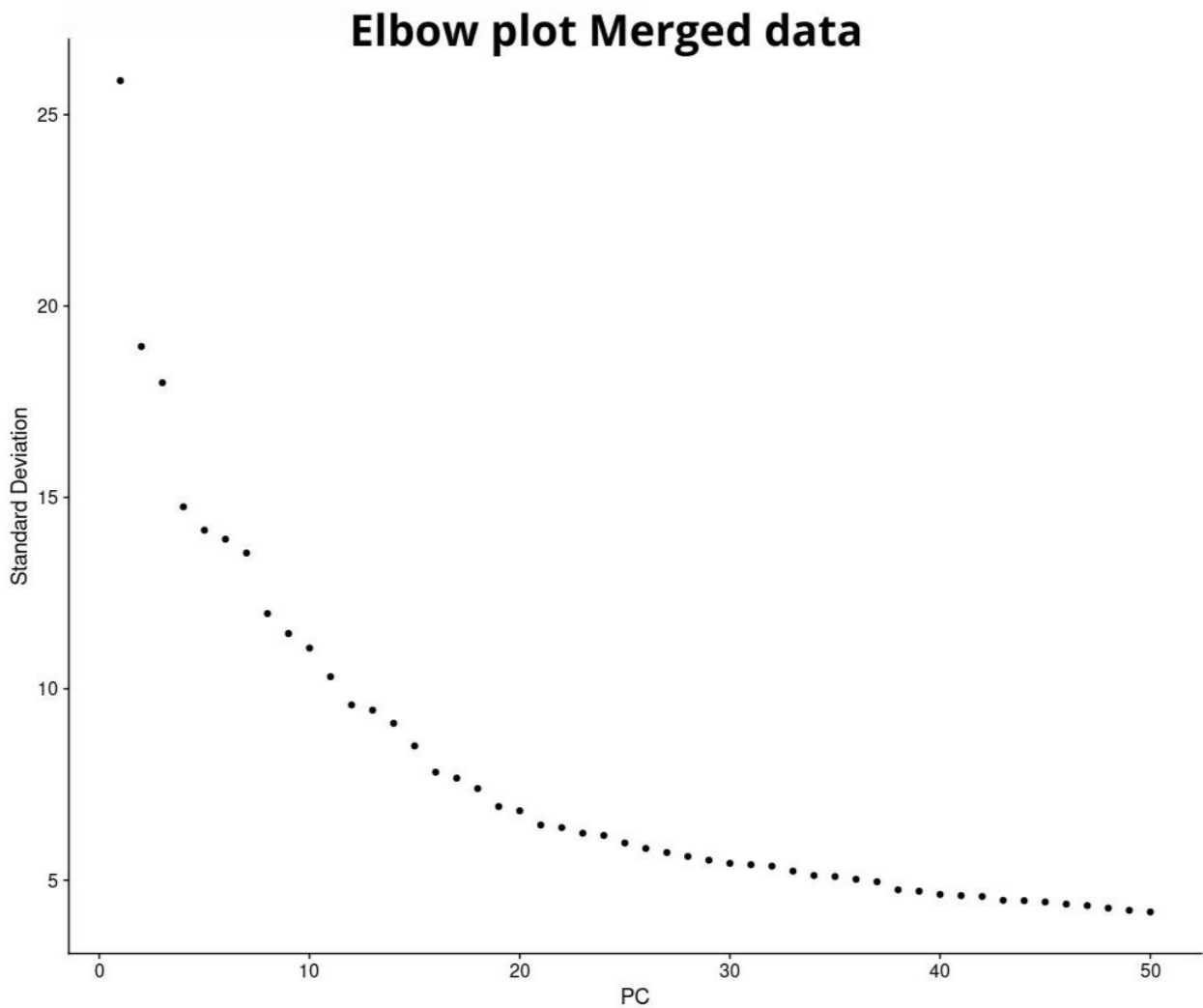



Figure B.5: Elbow plot Merged data

Código B.8: R studio code extract – Data non-linear reduction

```

1 #- UMAP Reduction -#
2 samplesMerge <- RunUMAP(samplesMerge, dims=1:40, verbose=F)
3
4 DimPlot(samplesMerge, reduction = "umap", group.by = 'Treatment', ←
  ↪ pt.size = 1.5) + ggtitle("Merged UMAP divided by Condition") ←
  ↪ + scale_color_manual(labels = c("Treated", "Untreated"), ←
  ↪ values = c("#00bfc4", "#F8766D"))
5
6 DimPlot(samplesMerge, reduction = "umap", group.by = 'orig.ident', ←
  ↪ pt.size = 1.5) + ggtitle("Merged UMAP divided by Sample") ←
  ↪ scale_color_manual(labels = c("Treated1", "Treated2", "←
  ↪ Untreated1", "Untreated2"), values = c("dodgerblue3", "←
  ↪ goldenrod1", "darkviolet", "lightseagreen"))

```

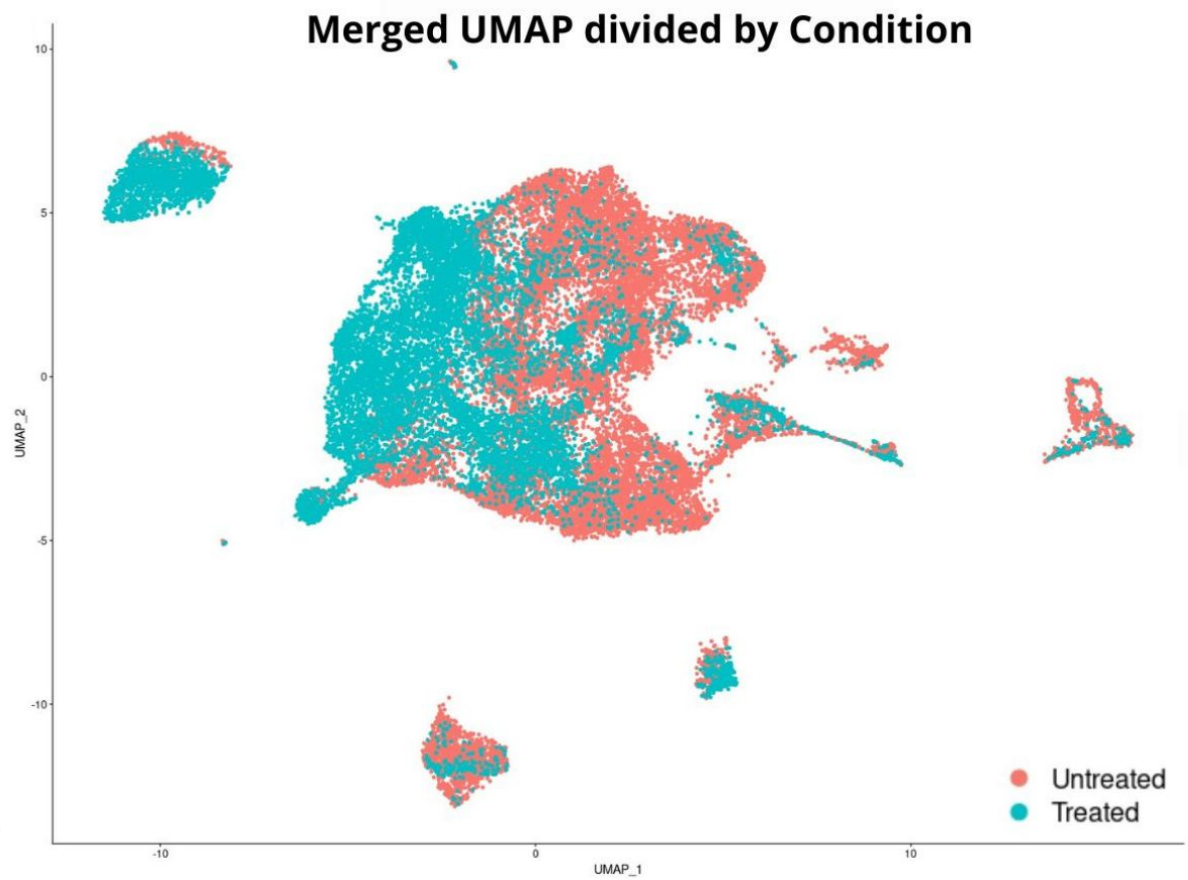


Figure B.6: UMAP of Merged data divided by Condition

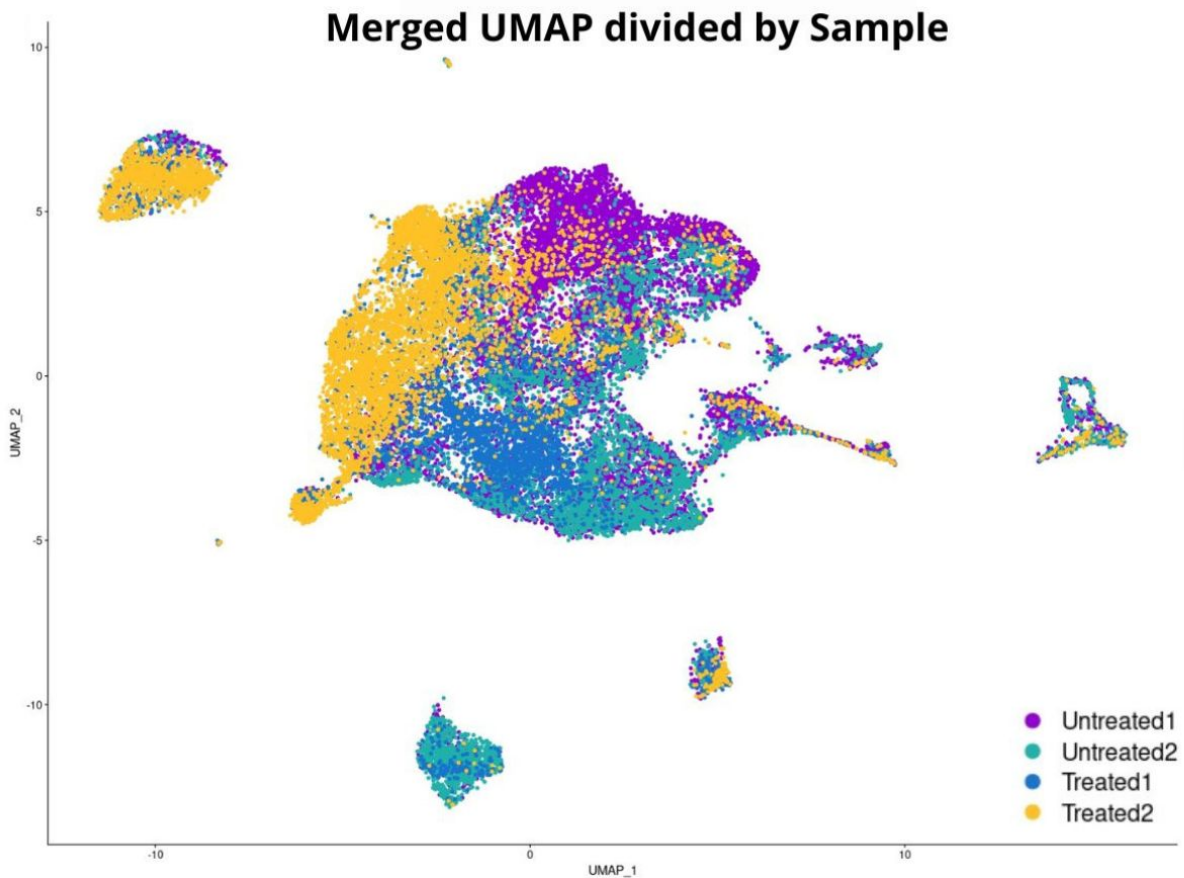


Figure B.7: UMAP of Merged data divided by Sample

Código B.9: R studio code extract – Samples Integration

```

1 #--- Re-Integration ---#
2
3 samplesSample.list <- list(Untreated1, Untreated2, Treated1, ←
  ↪ Treated2)
4
5 #- Normalice samples -#
6 sampleList <- list(Untreated1, Untreated2, Treated1, Treated2)
7 sampleList <- lapply(X = sampleList, FUN = function(x) {
8   x <- SCTransform(object = x, vars.to.regress = "percent.mt", ←
  ↪ verbose = FALSE, variable.features.n = 3000)
9 })
10
11 #- Select integration features and prepare for integration -#
12 features <- SelectIntegrationFeatures(object.list = sampleList)
13 sampleList <- PrepSCTIntegration(object.list = sampleList, anchor.←
  ↪ features = features)
14
15

```

```

16 #- Select integration anchors -#
17 data.anchors <- FindIntegrationAnchors(object.list = sampleList, ↵
    ↵ normalization.method = "SCT", anchor.features = features)
18 save(data.anchors, file=file.path("~/", "samplesInteg.anchors.rdata" ↵
    ↵ ))
19
20 #- Integrate -#
21 sample.integrated <- IntegrateData(anchorset = data.anchors, ↵
    ↵ normalization.method = "SCT")
22 saveRDS(sample.integrated, "sample.integrated.rds")

```

Código B.10: R studio code extract – Data linear reduction

```

1 #- PCA reduction -#
2 sample.integrated <- RunPCA(sample.integrated, npcs = 50, verbose=F ↵
    ↵ )
3
4 #Elbow plot
5 ElbowPlot(sample.integrated, ndims = 50, reduction = "pca") + ↵
    ↵ ggtitle("Elbow plot Integrated data")

```

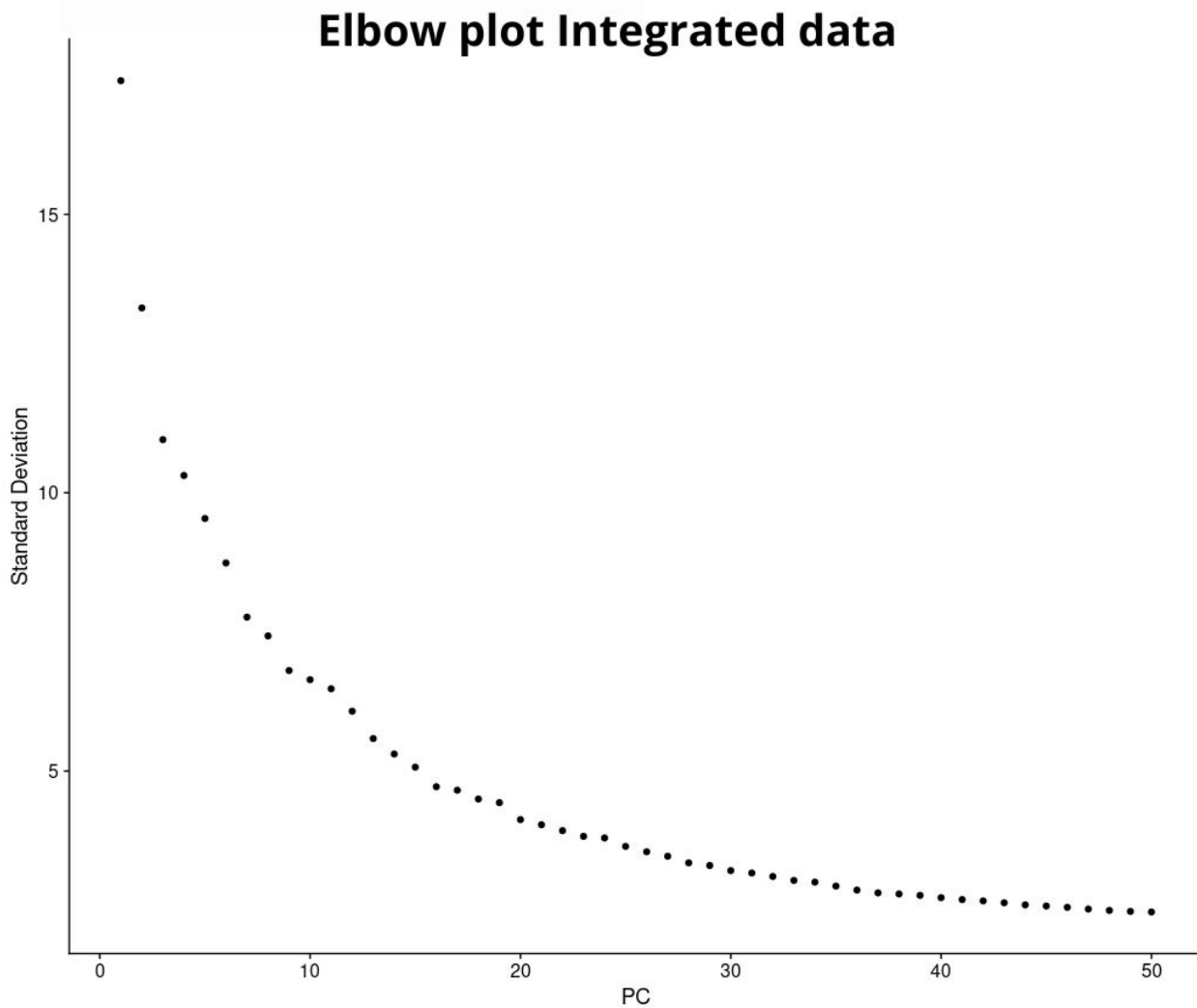


Figure B.8: Elbow plot Integrated data

Código B.11: R studio code extract – Data non-linear reduction

```

1 #- UMAP Reduction -#
2 sample.integrated <- RunUMAP(sample.integrated , dims=1:35, verbose=↵
  ↵ F)
3
4 DimPlot(sample.integrated , reduction = "umap" , group.by = 'orig.↵
  ↵ ident' , pt.size = 1.5, cols = my_colors) + ggtitle("↵
  ↵ Integrated UMAP divided by Sample") + scale_color_manual(↵
  ↵ labels = c("Untreated1", "Untreated2", "Treated1", "Treated2"↵
  ↵ ), values = my_colors)
5
6 DimPlot(sample.integrated , reduction = "umap" , group.by = '↵
  ↵ Treatment' , pt.size = 1.5) + ggtitle("Integrated UMAP ↵
  ↵ divided by Condition") + scale_color_manual(labels = c("↵

```

```
↪ Untreated", "Treated"), values = c("#F8766D", "#00bfc4"))
```

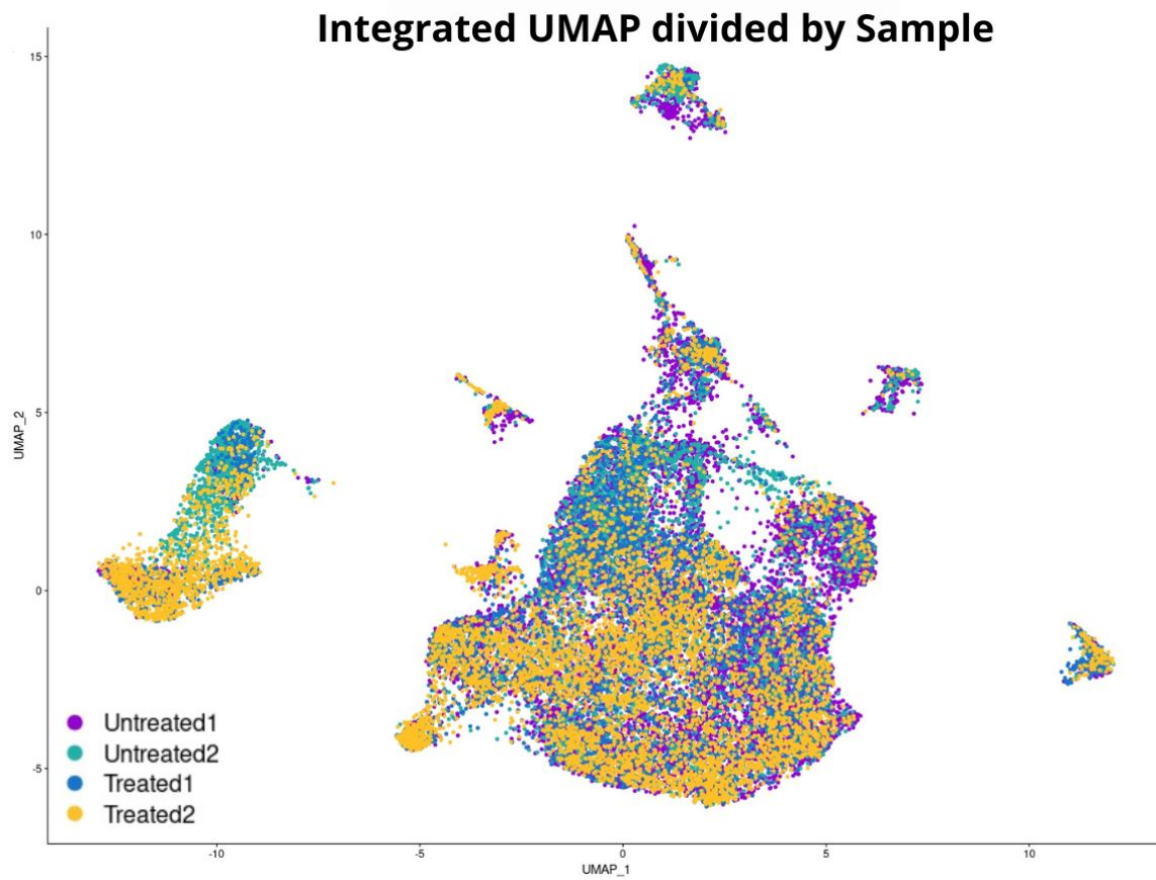


Figure B.9: UMAP of Integrated data divided by Sample

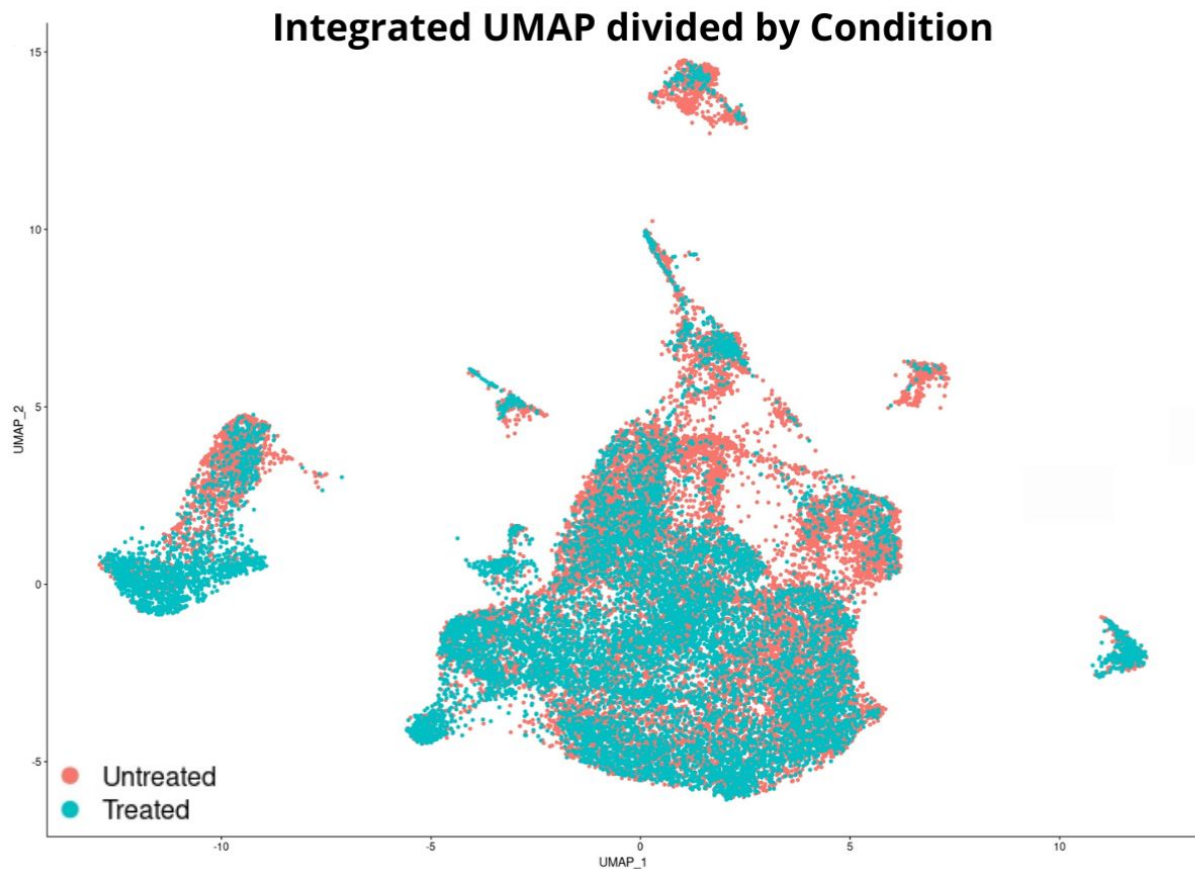


Figure B.10: UMAP of Integrated data divided by Condition

Código B.12: R studio code extract – Cells clustering

```

1 #--- Clustering ---#
2
3 DefaultAssay(object = sample.integrated) <- "integrated"
4 sample.integrated <- FindNeighbors(sample.integrated, dims = 1:35, ↵
  ↵ verbose = FALSE)
5 sample.integrated <- FindClusters(sample.integrated, resolution = ↵
  ↵ c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0))
6 clustree(sample.integrated)

```

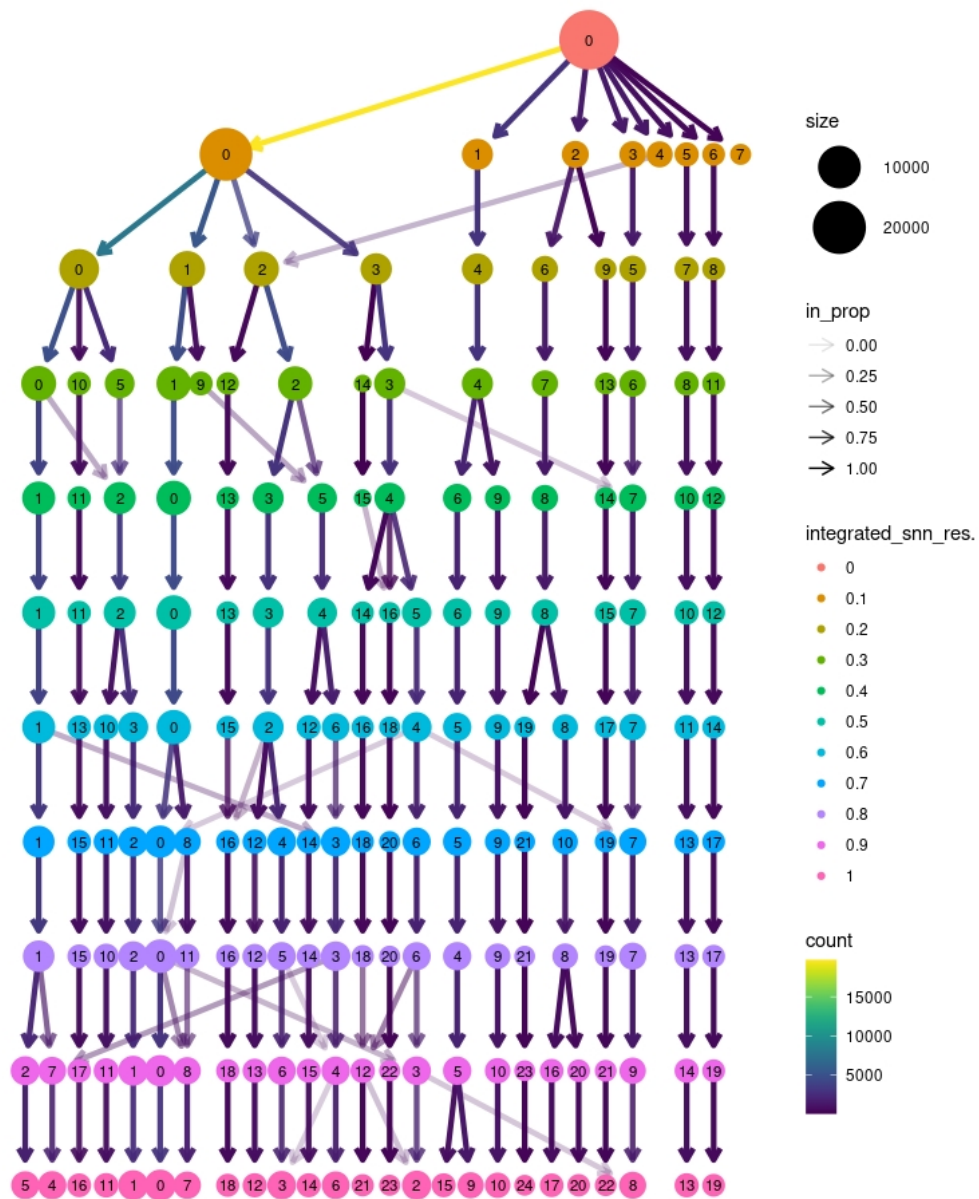


Figure B.11: Integrated data Clustree

Código B.13: R studio code extract – Contaminant Cd3e analysis

```

1
2 #- Chosen resolution -#
3 sample.integrated <- FindClusters(sample.integrated, resolution = ↵
  ↵ 0.1)
4 DimPlot(sample.integrated, reduction = "umap", pt.size = 1.5, cols↵
  ↵ = my_colors) + ggtitle('Integrated UMAP Resolution 0.1')
5
6 #- Cd3e expression analysis -#

```



```
7 DefaultAssay(object = sample.integrated) <- "SCT"
8 FeaturePlot(sample.integrated, features=c("Cd3e"), min.cutoff = 0, ←
  ↪ pt.size = 1.5, cols = c("royalblue", "brown2"))
9 VlnPlot(sample.integrated, features = c("Cd3e"), pt.size = 0, cols ←
  ↪ = my_colors)
```

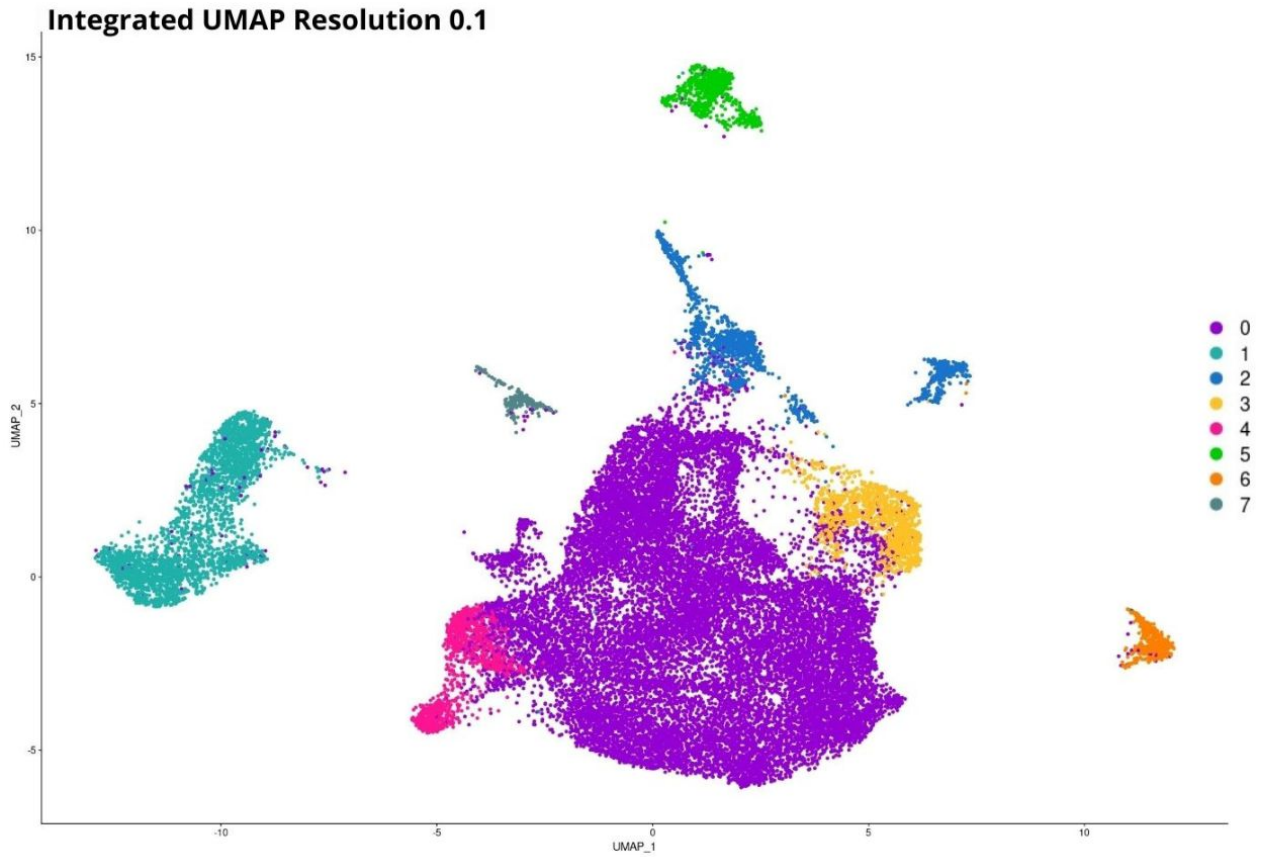


Figure B.12: Integrated data UMAP resolution 0.1

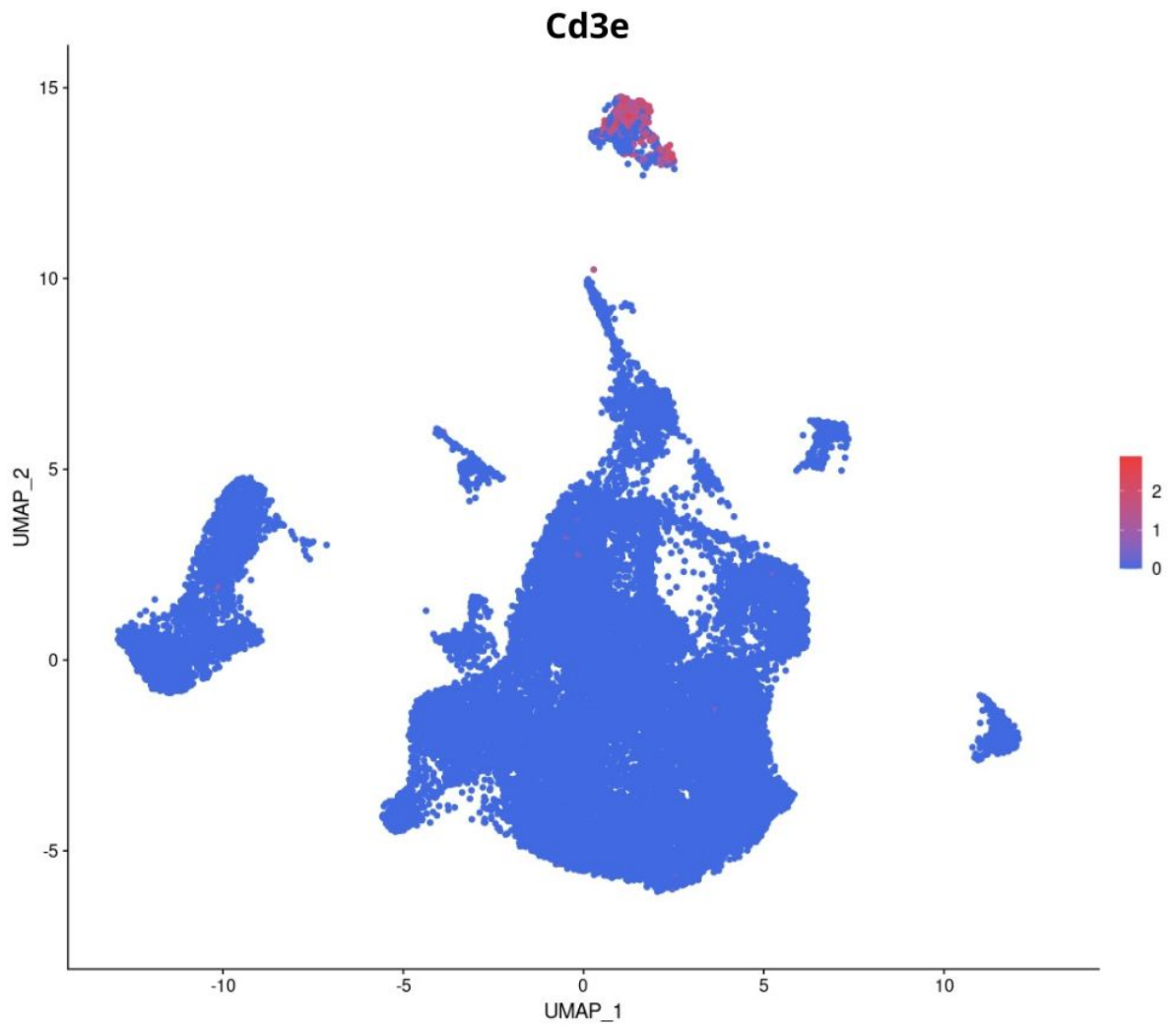


Figure B.13: Cd3e expression level

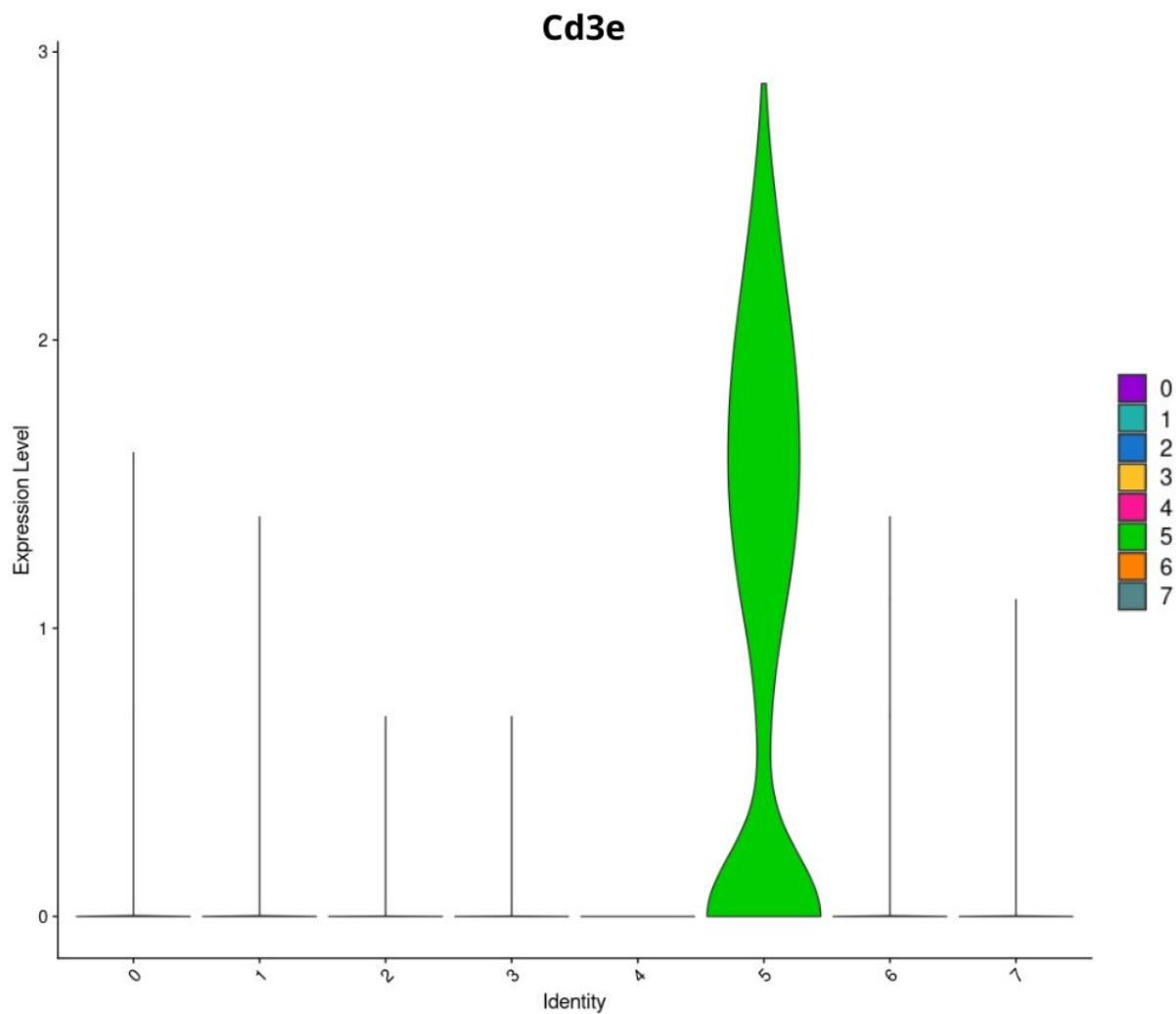


Figure B.14: Cd3e violin plot

Código B.14: R studio code extract – Extraction of contaminant cells and generation of new samples

```

1 #Extract contaminant cluster
2 sample.integrated <- subset(sample.integrated, idents = 5, invert <-
  <- = TRUE)
3
4 #- Save new samples -#
5 sampleList <- SplitObject(sample.integrated, split.by = "orig.<-
  <- ident")
6
7 Untreated1 <- sampleList[[1]]
8 saveRDS(Untreated1, "Untreated1V1Flt.rds")
9
10 Untreated2 <- sampleList[[2]]
11 saveRDS(Untreated2, "Untreated2V1Flt.rds")

```

```

12
13 Treated1 <- sampleList[[3]]
14 saveRDS(Treated1, "Treated1V1Flt.rds")
15
16 Treated2 <- sampleList[[4]]
17 saveRDS(Treated2, "Treated2V1Flt.rds")

```

Código B.15: R studio code extract – Downsampling by

```

1 #-- Merging of samples by Condition --#
2
3 #- Merging Untreated samples -#
4 Merge_Untreated <- merge(Untreated1_filt , y=Untreated2_filt , add.cell.<-
  ↪ cell.ids = c("Untreated1_filt", "Untreated2_filt"), project ↪
  ↪ = "Untreated")
5 #number of cells:16.318
6
7 #- Merging of Treated samples -#
8 Merge_Treated <- merge(Treated1_filt , y=Treated2_filt , add.cell.<-
  ↪ ids = c("Treated1_filt", "Treated2_filt"), project = "↪
  ↪ Treated")
9 #number of cells:10.985
10
11 #Downsampling:
12 slctCellsUntreated <- sample(rownames(Merge_Untreated@meta.data), ↪
  ↪ 10985)
13 Merge_Untreated <- subset(Merge_Untreated, cells=↪
  ↪ slctCellsUntreated)
14
15 #Save
16 saveRDS(Merge_Untreated, "Merge_Untreated_V1.rds")
17 saveRDS(Merge_Treated, "Merge_Treated_V1.rds")
18
19
20 #- Merging Condition samples for integration -#
21 samplesMerge <- merge(Merge_Untreated, y=Merge_Treated, add.cell.<-
  ↪ ids=c("Merge_Untreated", y = "Merge_Treated"), project="↪
  ↪ Myeloids")

```

Código B.16: R studio code extract – Samples Integration

```

1 #--- Re-Integration ---#
2

```

```

3 #- Normalice samples -#
4 sampleList <- SplitObject(samplesMerge, split.by = "orig.ident")
5 sampleList <- lapply(X = sampleList, FUN = function(x) {
6   x <- SCTransform(object = x, vars.to.regress = "percent.mt", ←
   ↪ verbose = FALSE, variable.features.n = 3000)
7 })
8
9 #- Select integration features and prepare for integration -#
10 features <- SelectIntegrationFeatures(object.list = sampleList)
11 sampleList <- PrepSCTIntegration(object.list = sampleList, anchor.↵
   ↪ features = features)
12
13
14 #- Select integration anchors -#
15 data.anchors <- FindIntegrationAnchors(object.list = sampleList, ←
   ↪ normalization.method = "SCT", anchor.features = features)
16 save(data.anchors, file=file.path("~/", "samplesInteg.anchors.rdata↵
   ↪ "))
17
18 #- Integrate -#
19 sample.integrated <- IntegrateData(anchorset = data.anchors, ←
   ↪ normalization.method = "SCT")
20 saveRDS(sample.integrated, "sample.integrated.rds")

```

Código B.17: R studio code extract – Data linear reduction

```

1 #- PCA reduction -#
2 sample.integrated <- RunPCA(sample.integrated, npcs = 50, verbose=F↵
   ↪ )
3
4 #Elbow plot
5 ElbowPlot(sample.integrated, ndims = 50, reduction = "pca") + ↵
   ↪ ggtitle("Elbow plot Integrated data")

```

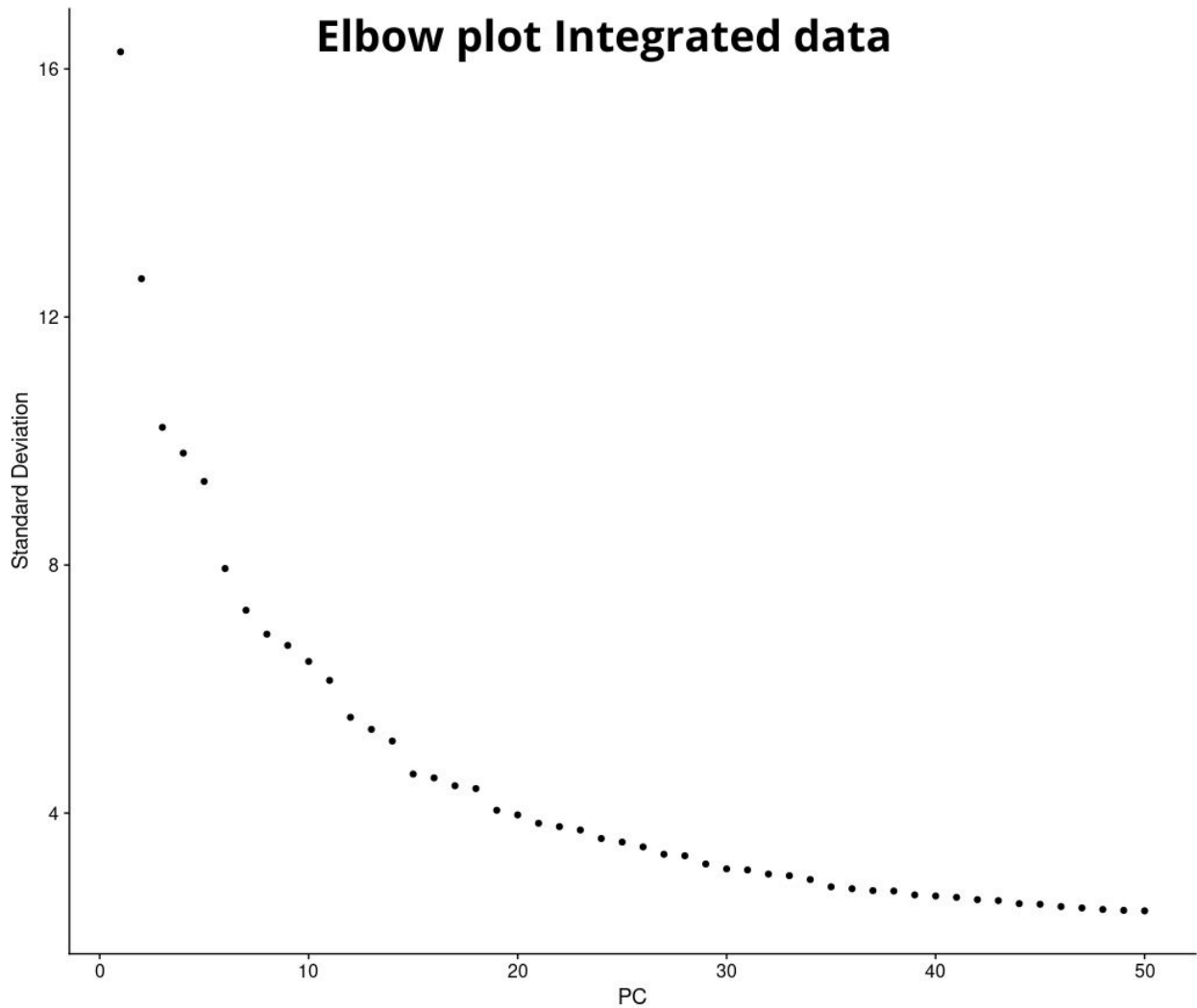


Figure B.15: Elbow plot Integrated data

Código B.18: R studio code extract – Data non-linear reduction

```

1 #- UMAP Reduction -#
2 sample.integrated <- RunUMAP(sample.integrated , dims=1:35, verbose=↵
  ↵ F)
3
4 DimPlot(sample.integrated , reduction = "umap" , group.by = 'orig.↵
  ↵ ident' , pt.size = 1.5, cols = my_colors) + ggtitle("↵
  ↵ Integrated UMAP divided by Sample") + scale_color_manual(↵
  ↵ labels = c("Untreated1", "Untreated2", "Treated1", "Treated2"↵
  ↵ ), values = my_colors)
5
6 DimPlot(sample.integrated , reduction = "umap" , group.by = '↵
  ↵ Treatment' , pt.size = 1.5) + ggtitle("Integrated UMAP ↵
  ↵ divided by Condition") + scale_color_manual(labels = c("↵

```

```
↪ Untreated", "Treated"), values = c("#F8766D", "#00bfc4"))
```

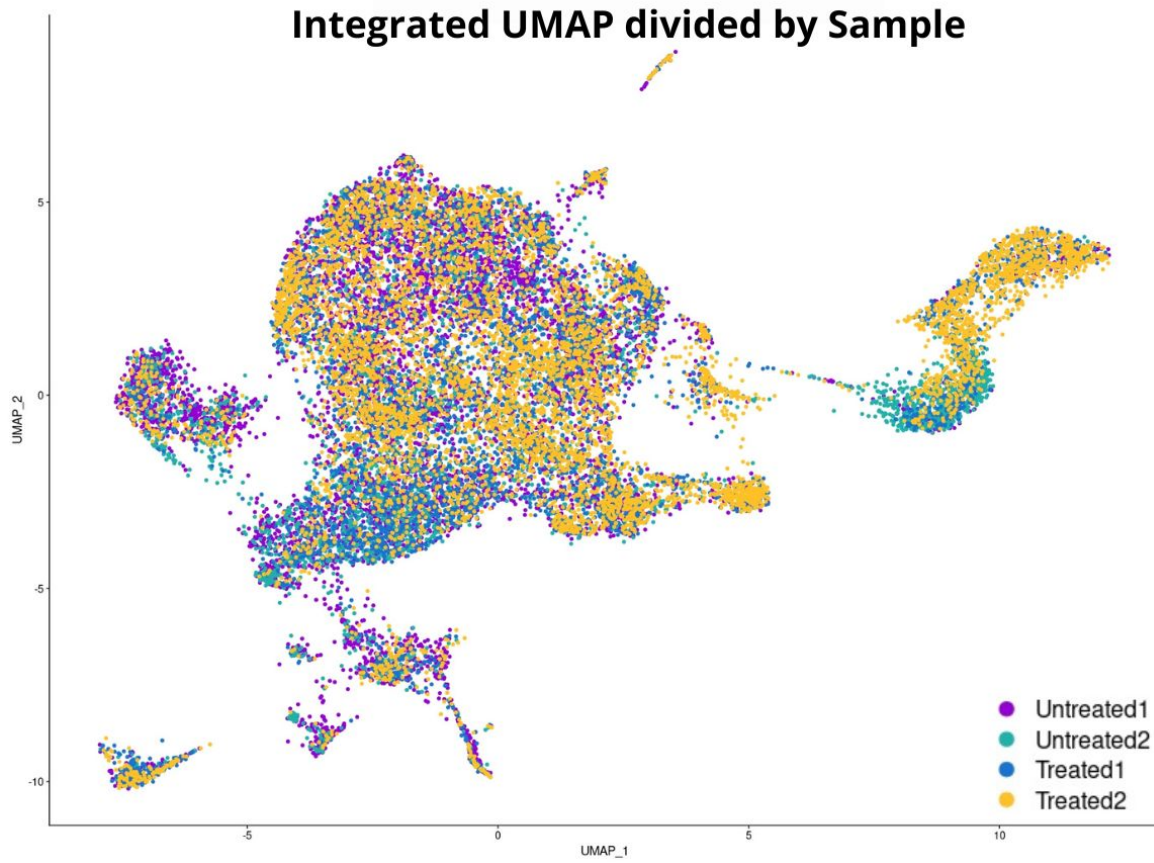


Figure B.16: UMAP of Integrated data divided by Sample

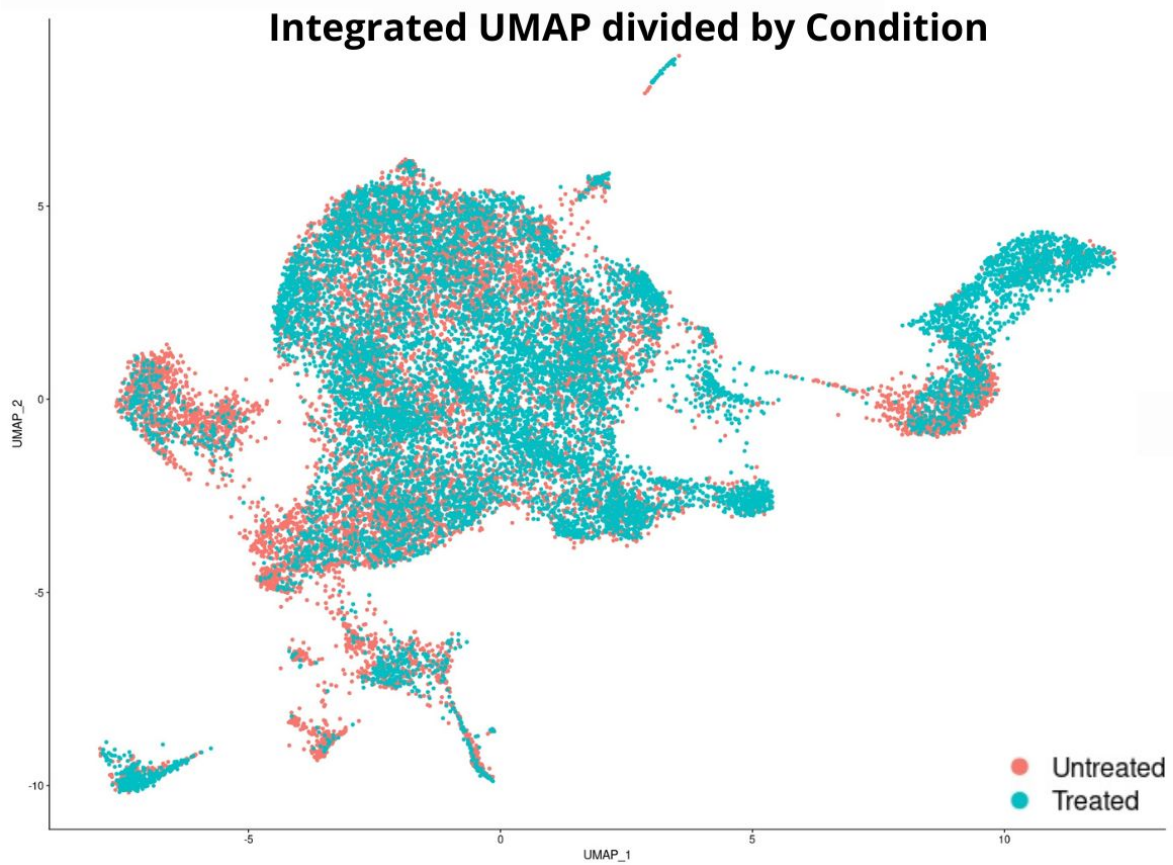


Figure B.17: UMAP of Integrated data divided by Condition

Código B.19: R studio code extract – Cells clustering

```

1 #--- Clustering ---#
2
3 DefaultAssay(object = sample.integrated) <- "integrated"
4 sample.integrated <- FindNeighbors(sample.integrated, dims = 1:25, ↵
  ↵ verbose = FALSE)
5 sample.integrated <- FindClusters(sample.integrated, resolution = ↵
  ↵ c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0))
6 clustree(sample.integrated)

```

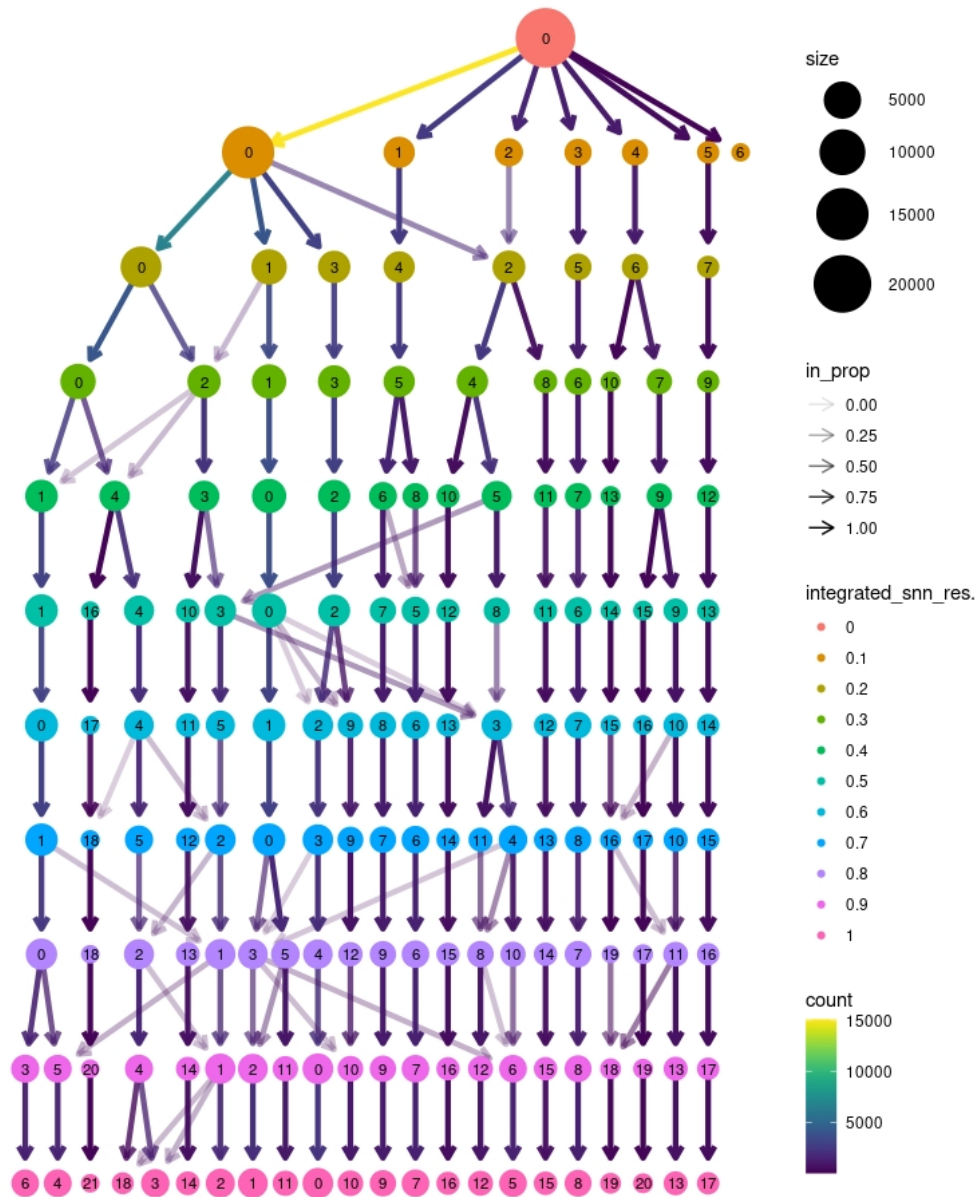



Figure B.18: Integrated data Clustree

Código B.20: R studio code extract – Myeloid marker Ptprc analysis

```

1 DefaultAssay(object = samplesIntegrated) <- "SCT"
2 FeaturePlot(samplesIntegrated, features=c("Ptprc"), min.cutoff = ↵
  ↵ 0, pt.size = 1, cols = c("royalblue", "brown2"))

```

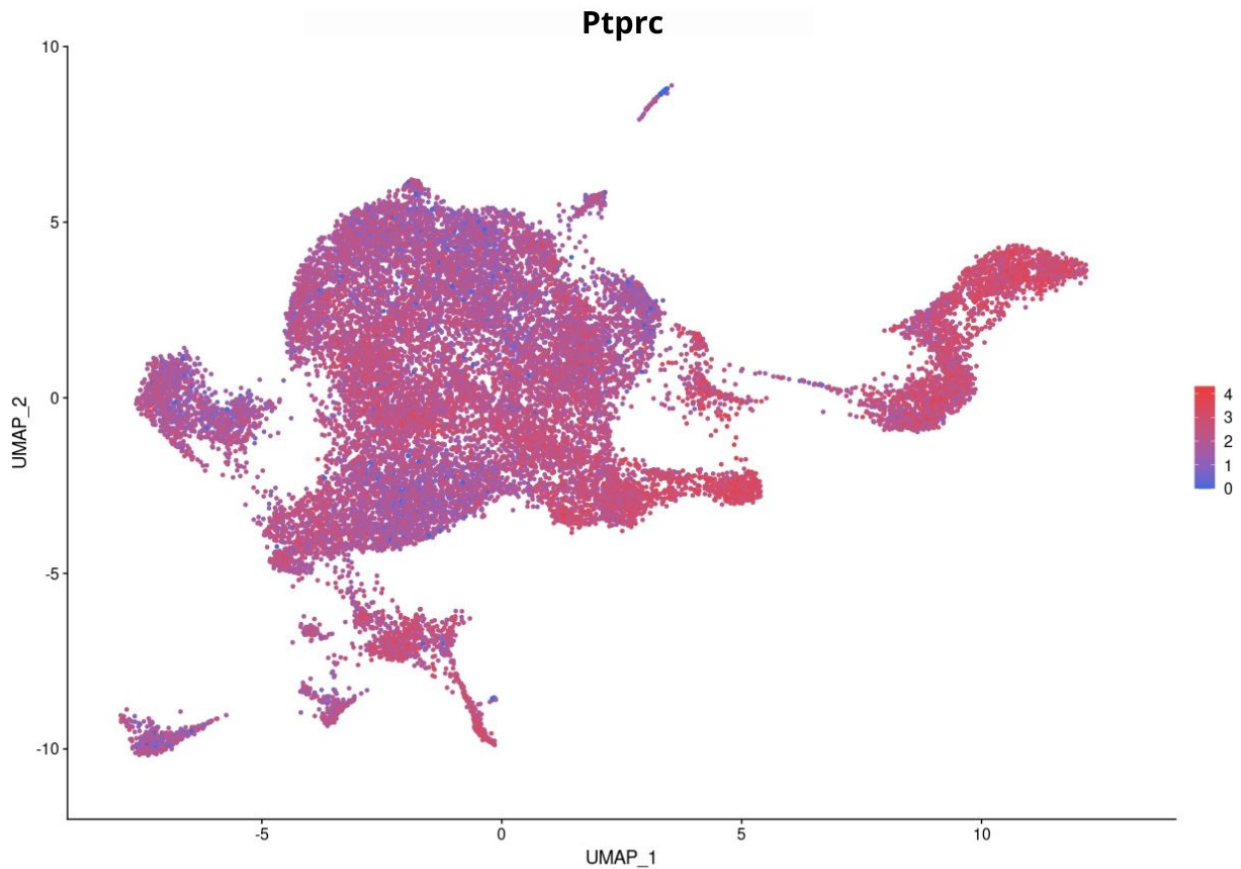


Figure B.19: Ptprc expression level

Código B.21: R studio code extract – Generation of higher resolution clusters to get Ptprc- cells isolated

```

1
2 DefaultAssay(object = samplesIntegrated) <- "integrated"
3 samplesIntegrated <- FindNeighbors(samplesIntegrated, dims = ↵
  ↵ 1:25, verbose = FALSE)
4
5 #- Chosen resolution -#
6 samplesIntegrated <- FindClusters(samplesIntegrated, resolution = ↵
  ↵ 6.9)
7 DimPlot(samplesIntegrated, reduction = "umap", label = TRUE, pt.↵
  ↵ size = 1) + ggtitle('Resolution 6.9')
8
9 #- Ptprc check -#
10 DefaultAssay(object = samplesIntegrated) <- "SCT"
11 VlnPlot(samplesIntegrated, features = c("Ptprc"), pt.size = 0)
12
13 #Contaminant cluster: 80

```

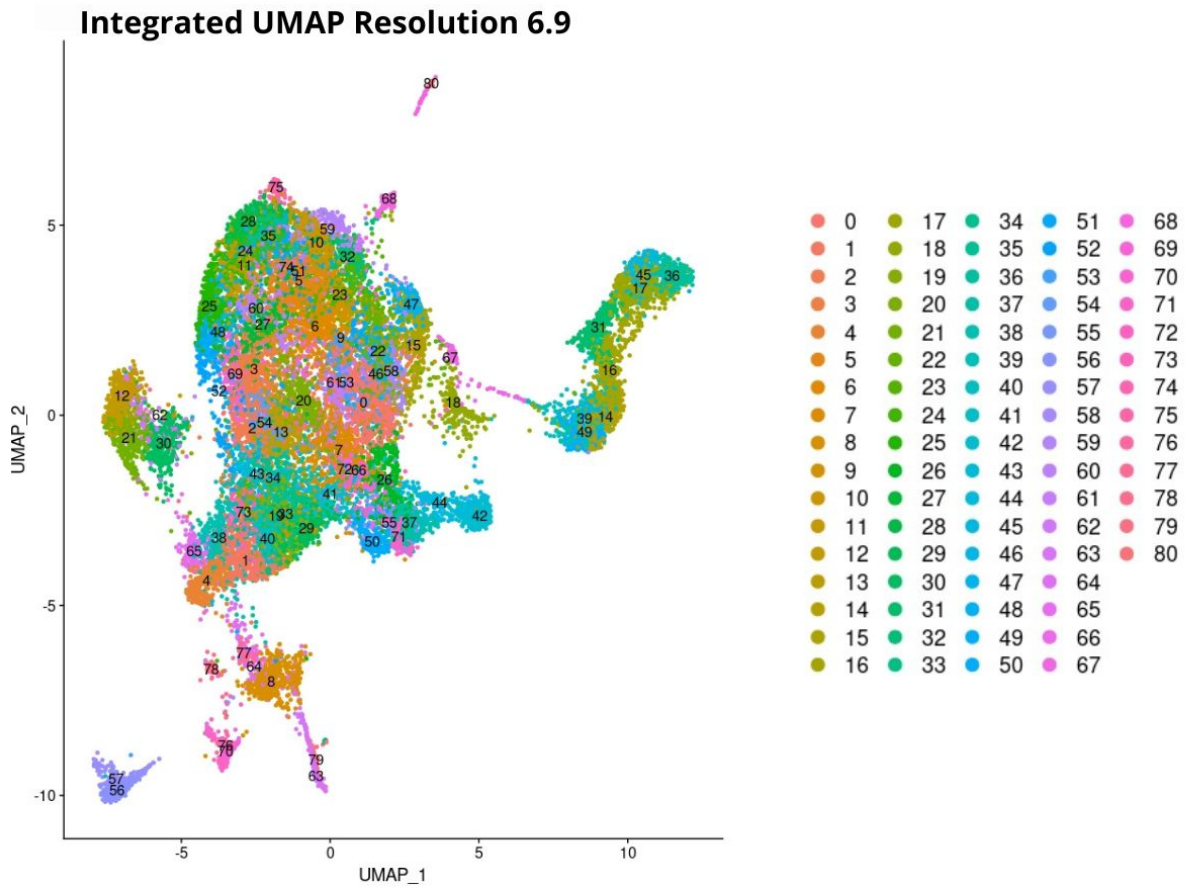


Figure B.20: Integrated data UMAP resolution 6.9

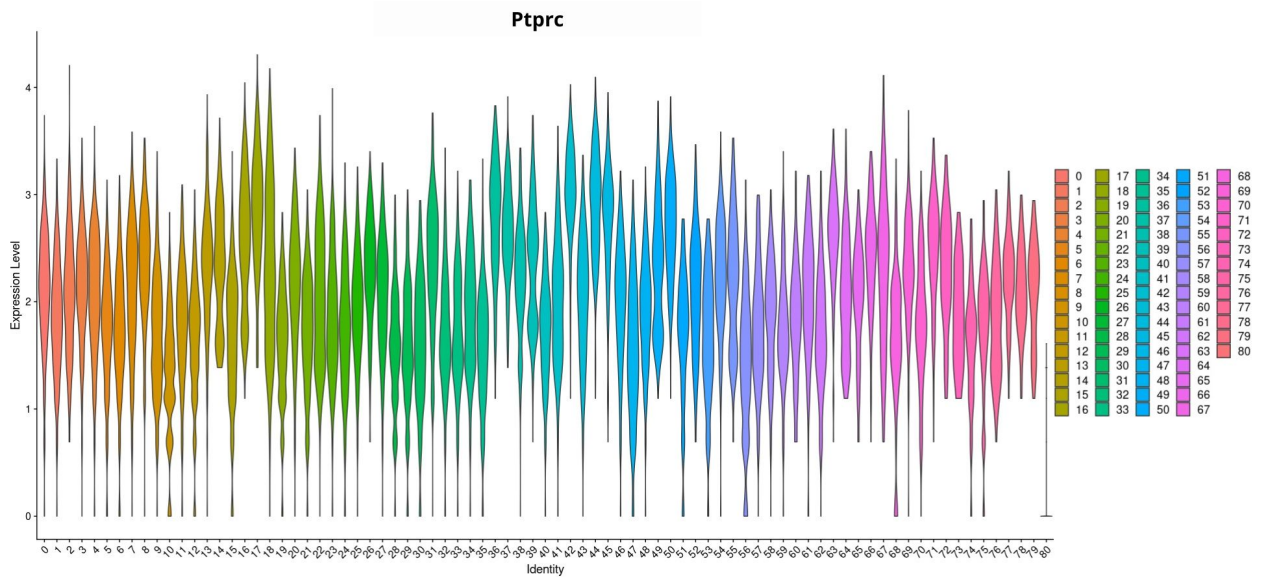


Figure B.21: Ptprc violin plot

Código B.22: R studio code extract – Extraction of contaminant cells and generation of new samples

```
1 #Extract contaminant cluster
2 sample.integrated <- subset(sample.integrated , idents = 80, invert←
  ↪ = TRUE)
3
4 #- Save new samples -#
5 sampleList <- SplitObject(sample.integrated , split.by = "orig.←
  ↪ ident")
6
7 Untreated1 <- sampleList[[1]]
8 saveRDS(Untreated1 , "Untreated1V1Flt.rds")
9
10 Untreated2 <- sampleList[[2]]
11 saveRDS(Untreated2 , "Untreated2V1Flt.rds")
12
13 Treated1 <- sampleList[[3]]
14 saveRDS(Treated1 , "Treated1V1Flt.rds")
15
16 Treated2 <- sampleList[[4]]
17 saveRDS(Treated2 , "Treated2V1Flt.rds")
```

Código B.23: R studio code extract – Downsampling by

```
1 #-- Merging of samples by Condition --#
2
3 #- Merging Untreated samples -#
4 Merge_Untreated <- merge(Untreated1_filt , y=Untreated2_filt , add.←
  ↪ cell.ids = c("Untreated1_filt" , "Untreated2_filt") , project ←
  ↪ = "Untreated")
5 #number of cells:10.974
6
7 #- Merging of Treated samples -#
8 Merge_Treated <- merge(Treated1_filt , y=Treated2_filt , add.cell.←
  ↪ ids = c("Treated1_filt" , "Treated2_filt") , project = "←
  ↪ Treated")
9 #number of cells:10.968
10
11 #Downsampling:
12 slctCellsUntreated <- sample(rownames(Merge_Untreated@meta.data) , ←
  ↪ 10.968)
13 Merge_Untreated <- subset(Merge_Untreated , cells=←
  ↪ slctCellsUntreated)
14
```

```

15 #Save
16 saveRDS(Merge_Untreated, "Merge_Untreated_V2.rds")
17 saveRDS(Merge_Treated, "Merge_Treated_V2.rds")
18
19
20 #- Merging Condition samples for integration -#
21 samplesMerge <- merge(Merge_Untreated, y=Merge_Treated, add.cell.<-
  ↪ ids=c("Merge_Untreated", y = "Merge_Treated"), project="↪
  ↪ Myeloids")

```

Código B.24: R studio code extract – Samples Integration

```

1 #--- Re-Integration ---#
2
3 #- Normalice samples -#
4 sampleList <- SplitObject(samplesMerge, split.by = "orig.ident")
5 sampleList <- lapply(X = sampleList, FUN = function(x) {
6   x <- SCTransform(object = x, vars.to.regress = "percent.mt", ↪
  ↪ verbose = FALSE, variable.features.n = 5000)
7 })
8
9 #- Select integration features and prepare for integration -#
10 features <- SelectIntegrationFeatures(object.list = sampleList)
11 sampleList <- PrepSCTIntegration(object.list = sampleList, anchor.<-
  ↪ features = features)
12
13
14 #- Select integration anchors -#
15 data.anchors <- FindIntegrationAnchors(object.list = sampleList, ↪
  ↪ normalization.method = "SCT", anchor.features = features)
16 save(data.anchors, file=file.path("~/", "samplesInteg.anchors.rdata↪
  ↪ "))
17
18 #- Integrate -#
19 sample.integrated <- IntegrateData(anchorset = data.anchors, ↪
  ↪ normalization.method = "SCT")
20 saveRDS(sample.integrated, "sample.integrated.rds")

```

Código B.25: R studio code extract – Data linear reduction

```

1 #- PCA reduction -#
2 sample.integrated <- RunPCA(sample.integrated, npcs = 50, verbose=F↪
  ↪ )

```

3

```
4 #Elbow plot
```

```
5 ElbowPlot(sample.integrated, ndims = 50, reduction = "pca") +  
  ↪ ggtitle("Elbow plot Integrated data")
```

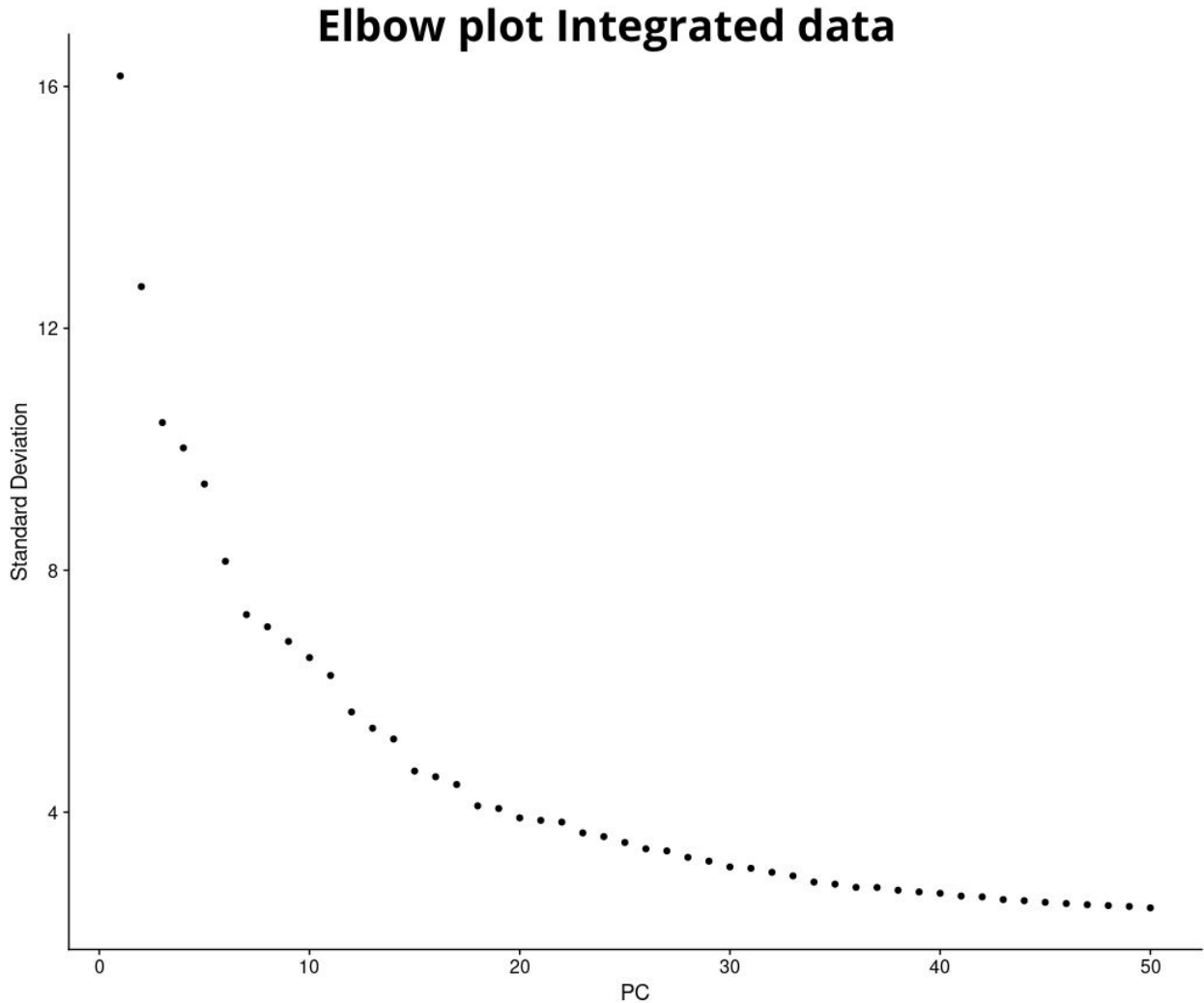


Figure B.22: Elbow plot Integrated data

Código B.26: R studio code extract – Data non-linear reduction

```
1 #- UMAP Reduction -#
```

```
2 sample.integrated <- RunUMAP(sample.integrated, dims=1:35, verbose=↪  
  ↪ F)
```

3

```
4 DimPlot(sample.integrated, reduction = "umap", group.by = 'orig.↪  
  ↪ ident', pt.size = 1.5, cols = my_colors) + ggtitle("↪  
  ↪ Integrated UMAP divided by Sample") + scale_color_manual(↪
```



```

↪ labels = c("Untreated1", "Untreated2", "Treated1", "Treated2" ↪
↪ ), values = my_colors)
5
6 DimPlot(sample.integrated, reduction = "umap", group.by = '↪
↪ Treatment', pt.size = 1.5) + ggtitle("Integrated UMAP ↪
↪ divided by Condition") + scale_color_manual(labels = c("↪
↪ Untreated", "Treated"), values = c("#F8766D", "#00bfc4"))

```

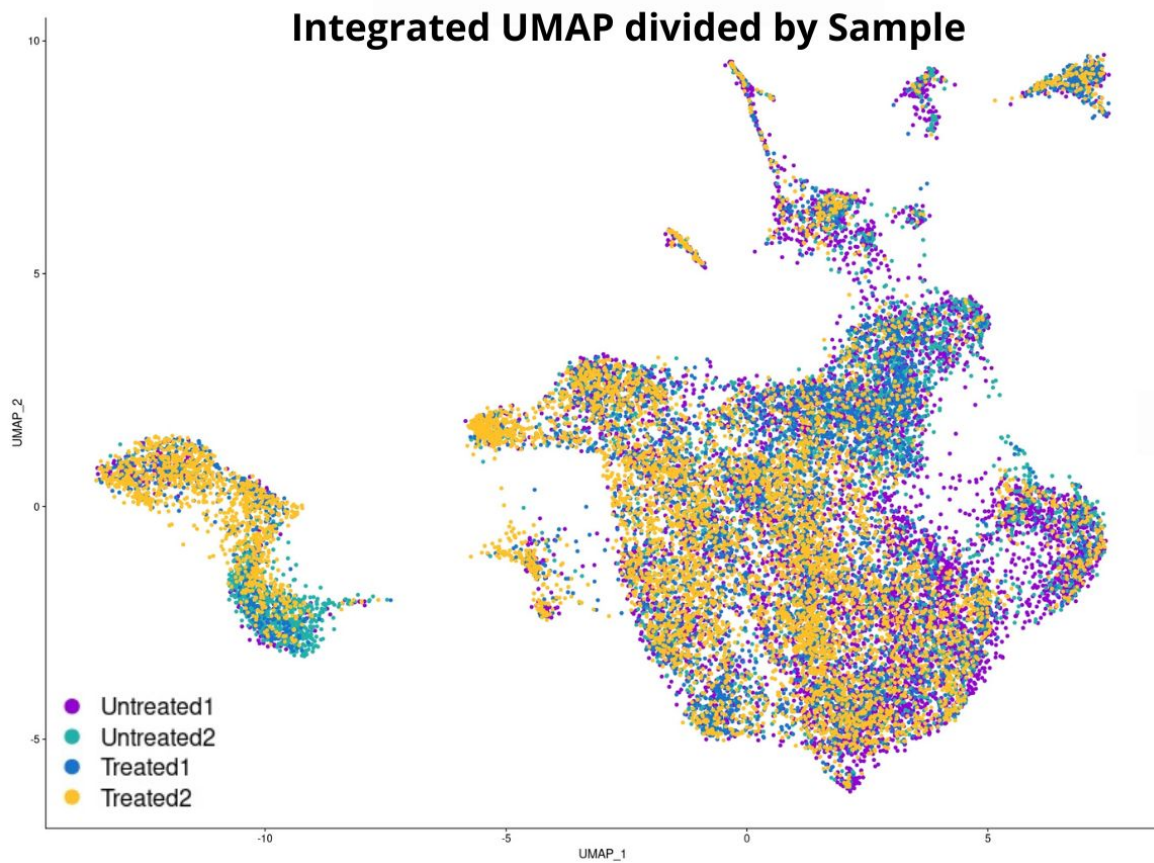


Figure B.23: UMAP of Integrated data divided by Sample

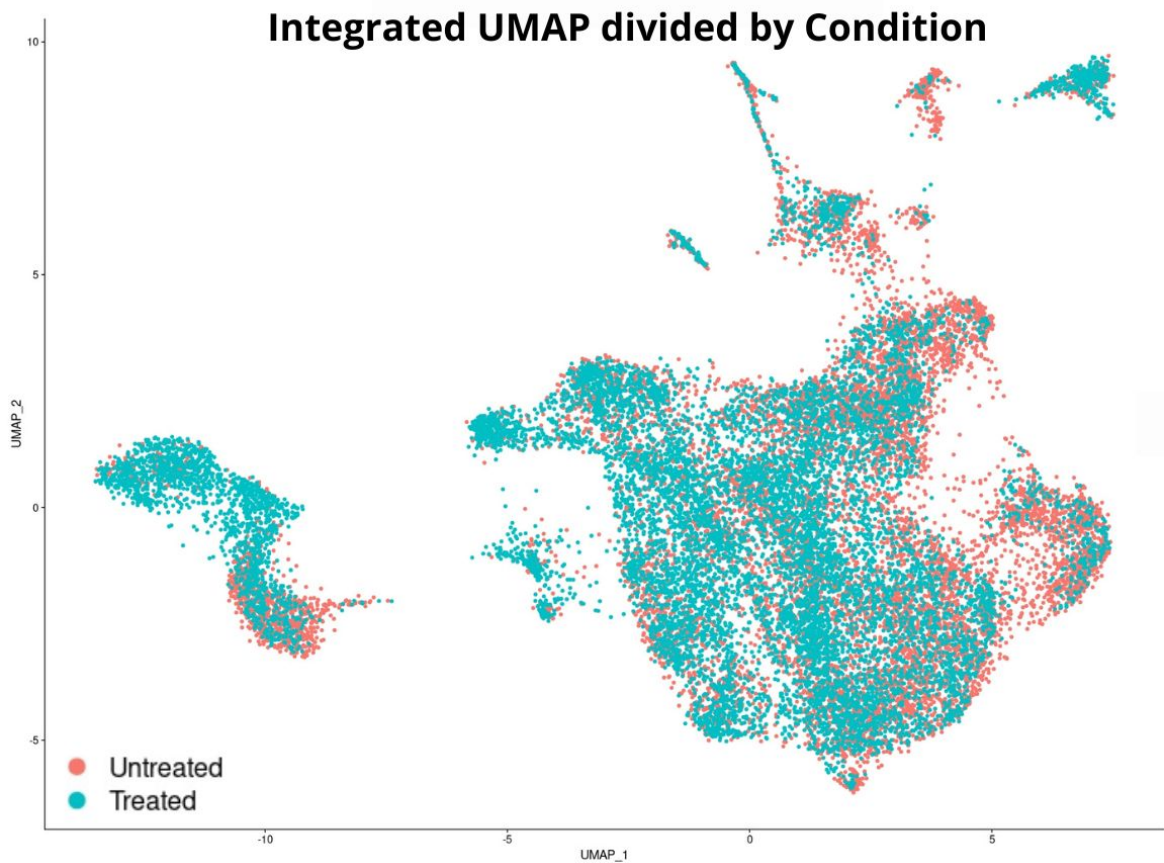


Figure B.24: UMAP of Integrated data divided by Condition

Código B.27: R studio code extract – Key genes analysis

```

1 DefaultAssay(object = samplesIntegrated) <- "SCT"
2
3 #- Granulocytes: -Mmp9
4 FeaturePlot(samplesIntegrated, features=c("Mmp9"), min.cutoff = 0,↵
  ↵ pt.size = 1.5, cols = c("royalblue", "brown2"))
5
6 #- Mono/Macro: Csf1r
7 FeaturePlot(samplesIntegrated, features=c("Csf1r"), min.cutoff = ↵
  ↵ 0, pt.size = 1.5, cols = c("royalblue", "brown2"))
8
9 #- DCs: Flt3 -#
10 FeaturePlot(samplesIntegrated, features=c("Flt3"), min.cutoff = 0,↵
  ↵ pt.size = 1.5, cols = c("royalblue", "brown2"))

```

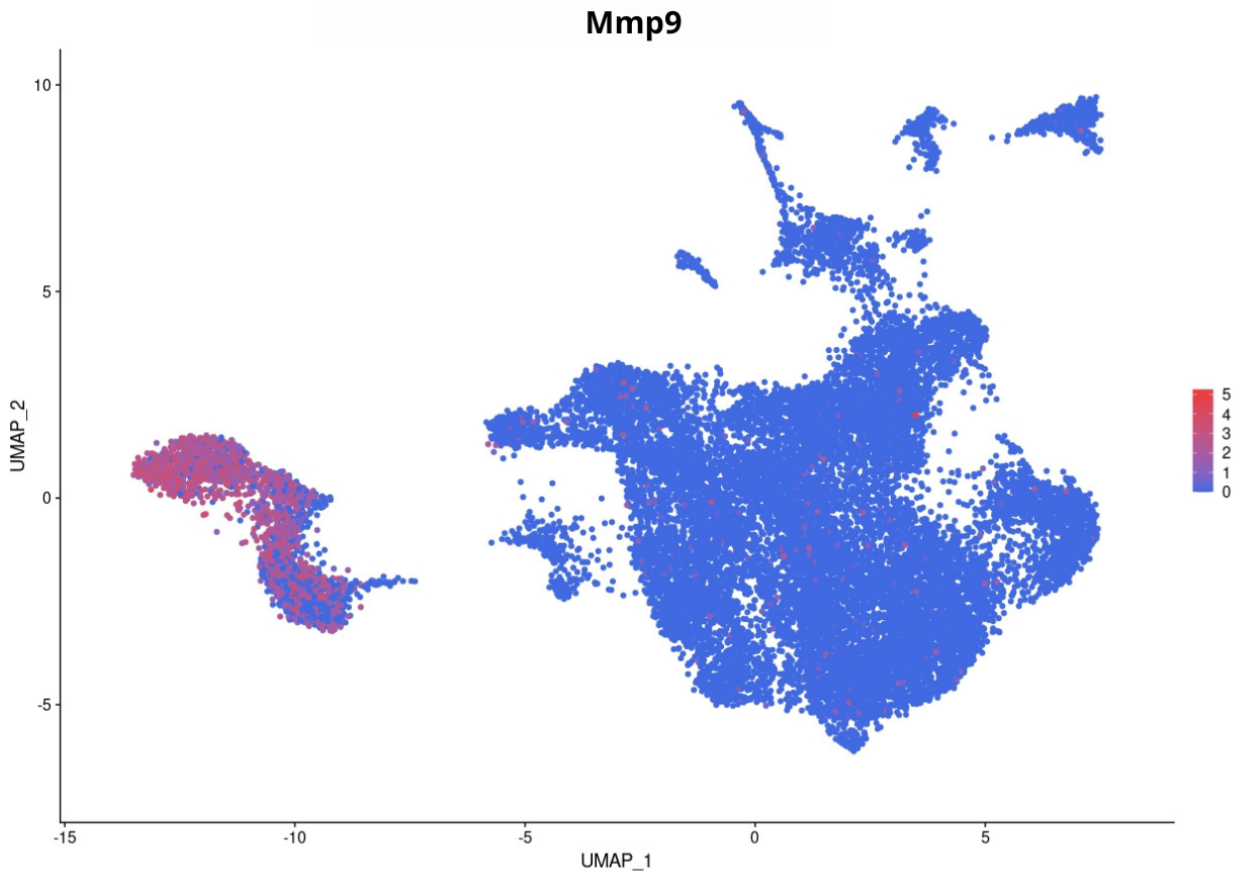



Figure B.25: Mmp9 expression level

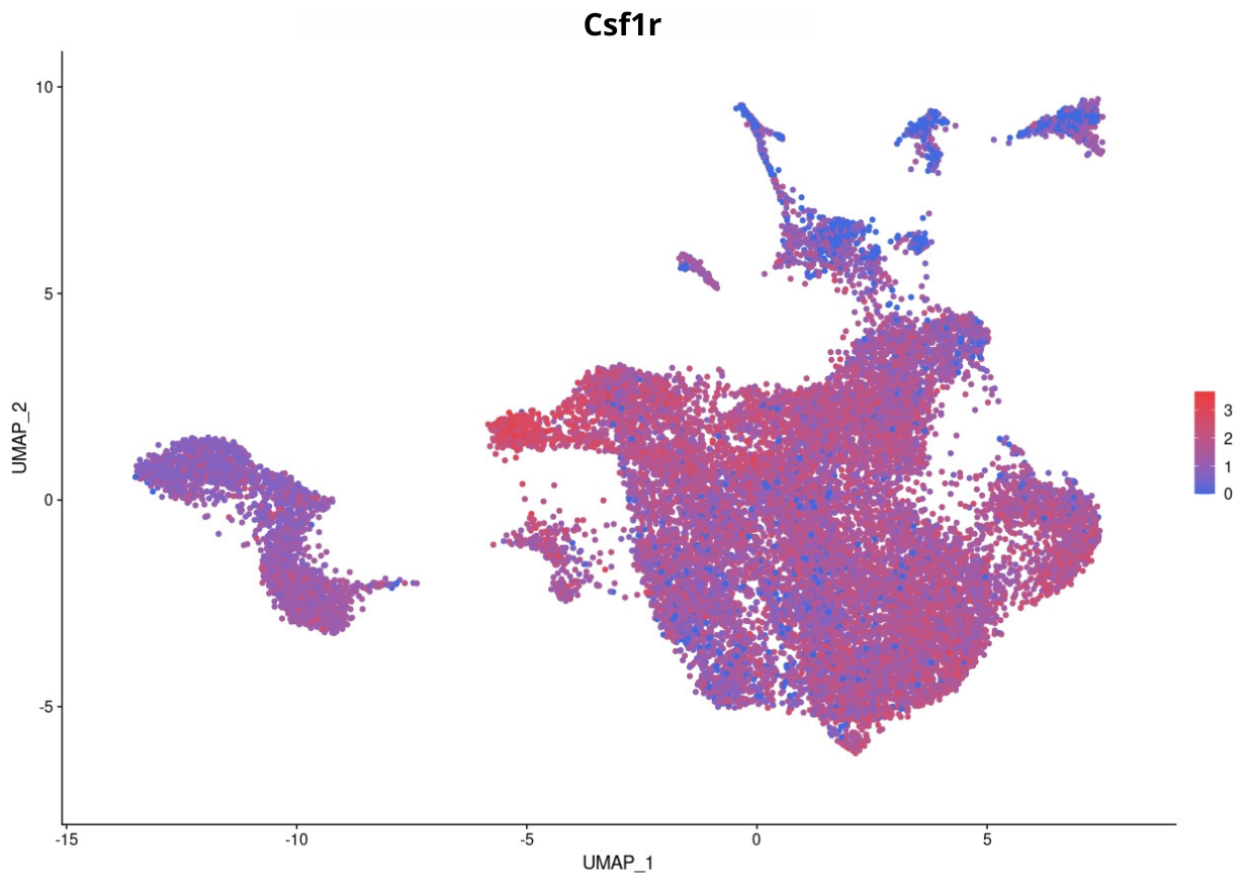


Figure B.26: Csf1r expression level

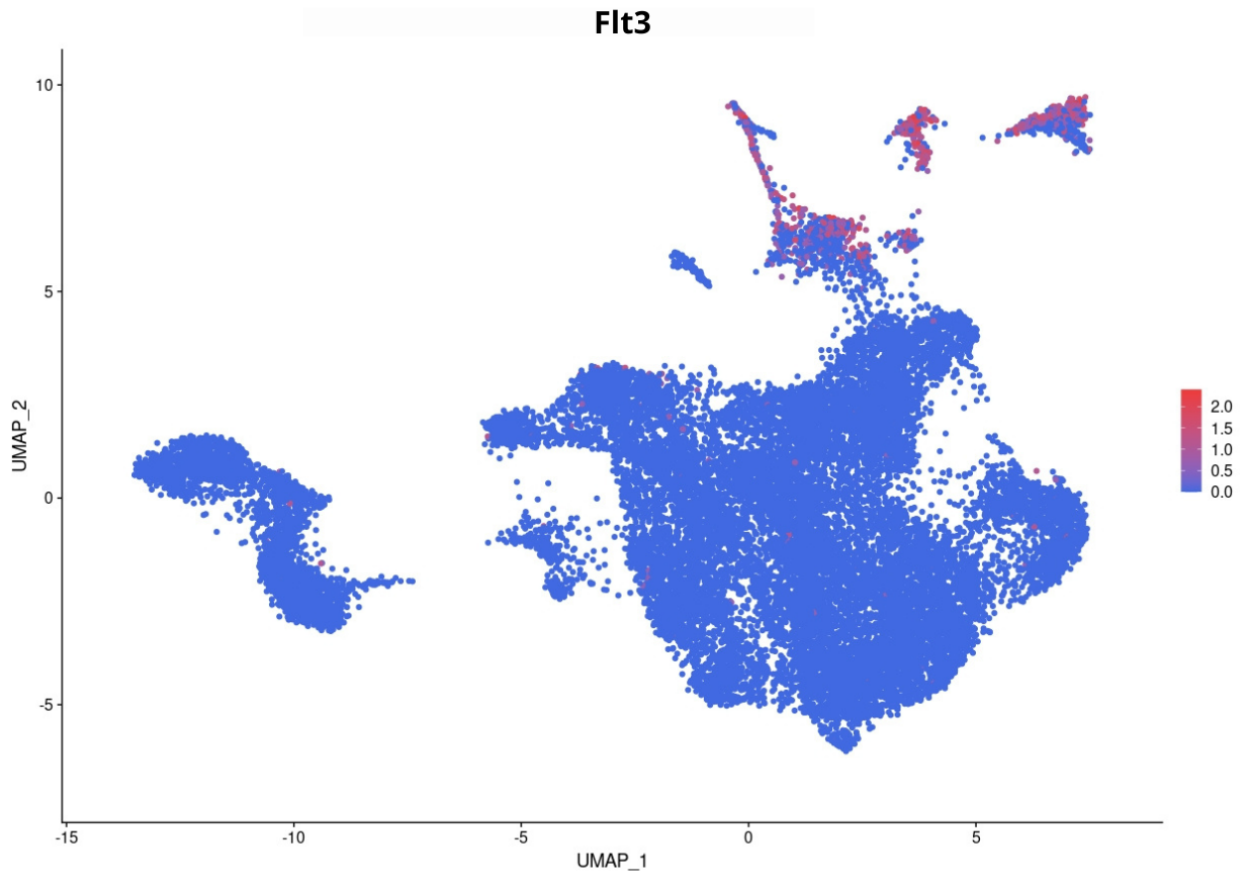


Figure B.27: Flt3 expression level

Código B.28: R studio code extract – Key genes analysis: Dotplot

```

1 DefaultAssay(object = samplesIntegrated) <- "integrated"
2
3 key_genes = c("Csf3r", "Mmp9", "Gata2", "Ccr3", "Cxcl2", "S100a8", ↵
  ↵ "S100a9", "Dusp1", "Xcr1", "Adam11", "Cd209d", "Spiib", "H2- ↵
  ↵ Oa", "Klrd1", "Bcl11a", "Flt3", "Ace", "Ccr2", "Lyz2", "Apoe ↵
  ↵ ", "Ctsb", "Mafb", "C1qa", "Ctss")
4 DotPlot(samplesIntegrated, features = key_genes, cols = c(" ↵
  ↵ royalblue", "brown2")) + RotatedAxis() + ggtitle('Myeloid ↵
  ↵ populations key genes')
```

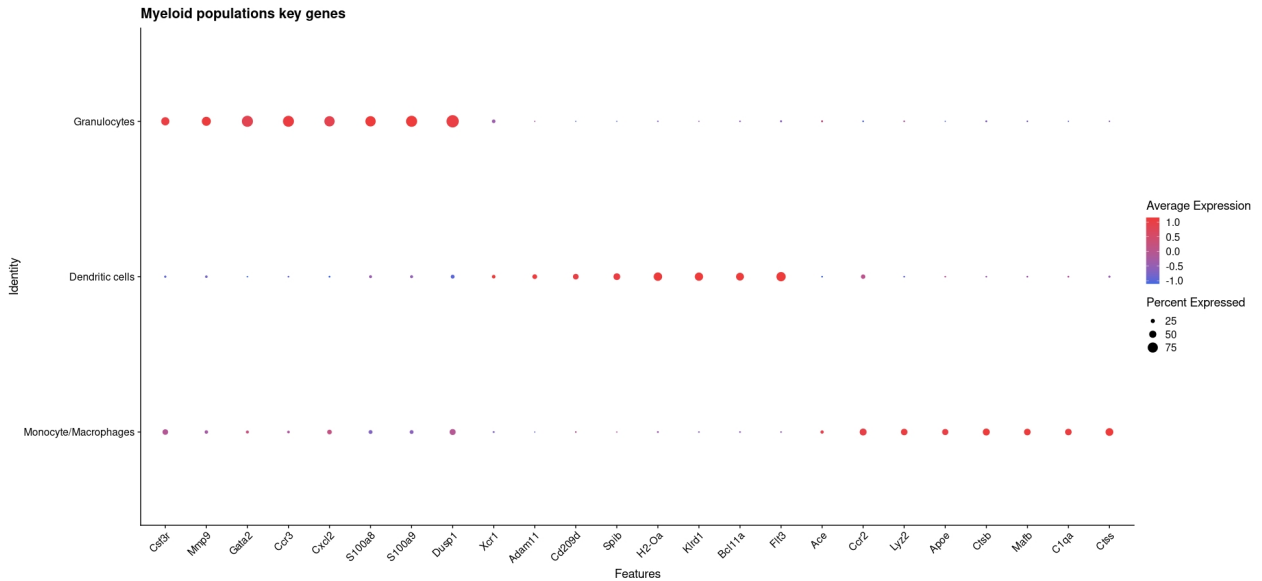


Figure B.28: Myeloid key genes Dotplot

Código B.29: R studio code extract – Manual selection: Granulocytes

```

1 ##--- Manual selection ---##
2 plot <- DimPlot(samplesIntegrated, reduction = "umap", label = ↵
  ↵ TRUE, pt.size = 1, cols = my_colors)
3 plot
4
5 #- Sell selection -#
6 select.cells <- CellSelector(plot = plot)
7 Idents(samplesIntegrated, cells = select.cells) <- "Granulocytes"
8 plot <- DimPlot(samplesIntegrated, reduction = "umap", label = ↵
  ↵ TRUE, pt.size = 1)
9
10 #- Subset -#
11 samplesIntegrated_Granulocytes <- subset(samplesIntegrated, idents↵
  ↵ = "Granulocytes")
12
13 DimPlot(samplesIntegrated_Granulocytes, reduction = "umap", pt.↵
  ↵ size = 1.5, cols = "dodgerblue3", group.by = "integrated_snn↵
  ↵ _res.0") + ggtitle('Granulocytes')
14
15 DimPlot(samplesIntegrated_Granulocytes, reduction = "umap", label ↵
  ↵ = FALSE, pt.size = 1.5, group.by = 'Treatment') + ggtitle('↵
  ↵ Granulocytes by Treatment')
16
17 #- Save data -#
18 saveRDS(samplesIntegrated_Granulocytes, "samplesIntegrated_↵

```

↳ GranulocytesSelect.rds")

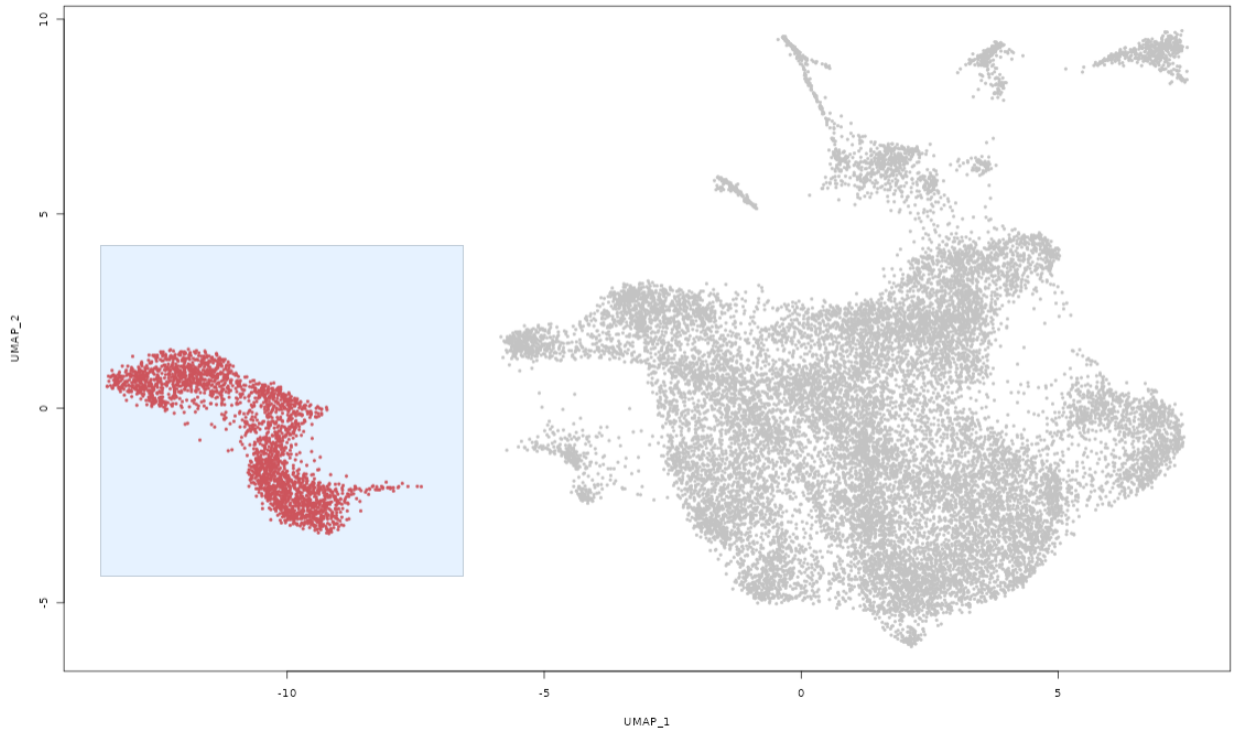


Figure B.29: Granulocytes manual selection

Granulocytes

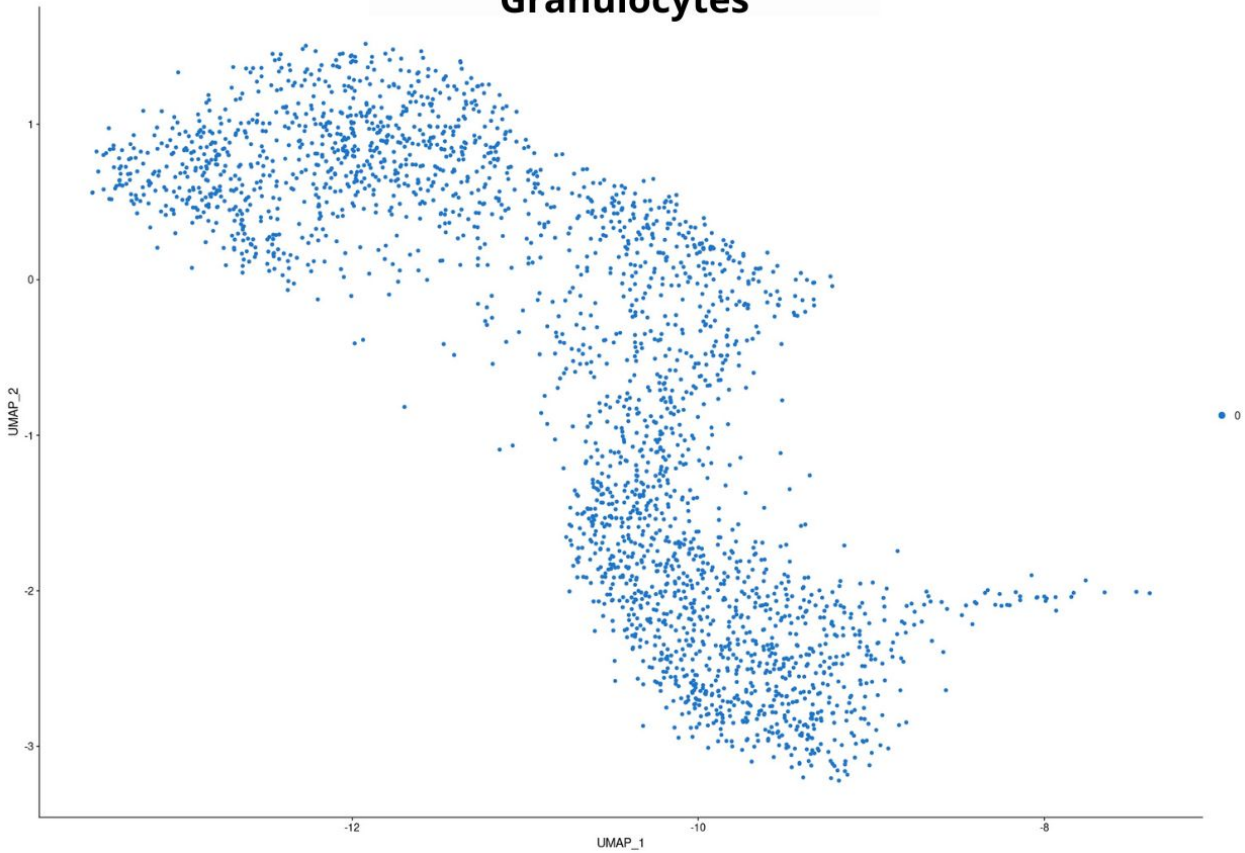


Figure B.30: Granulocytes

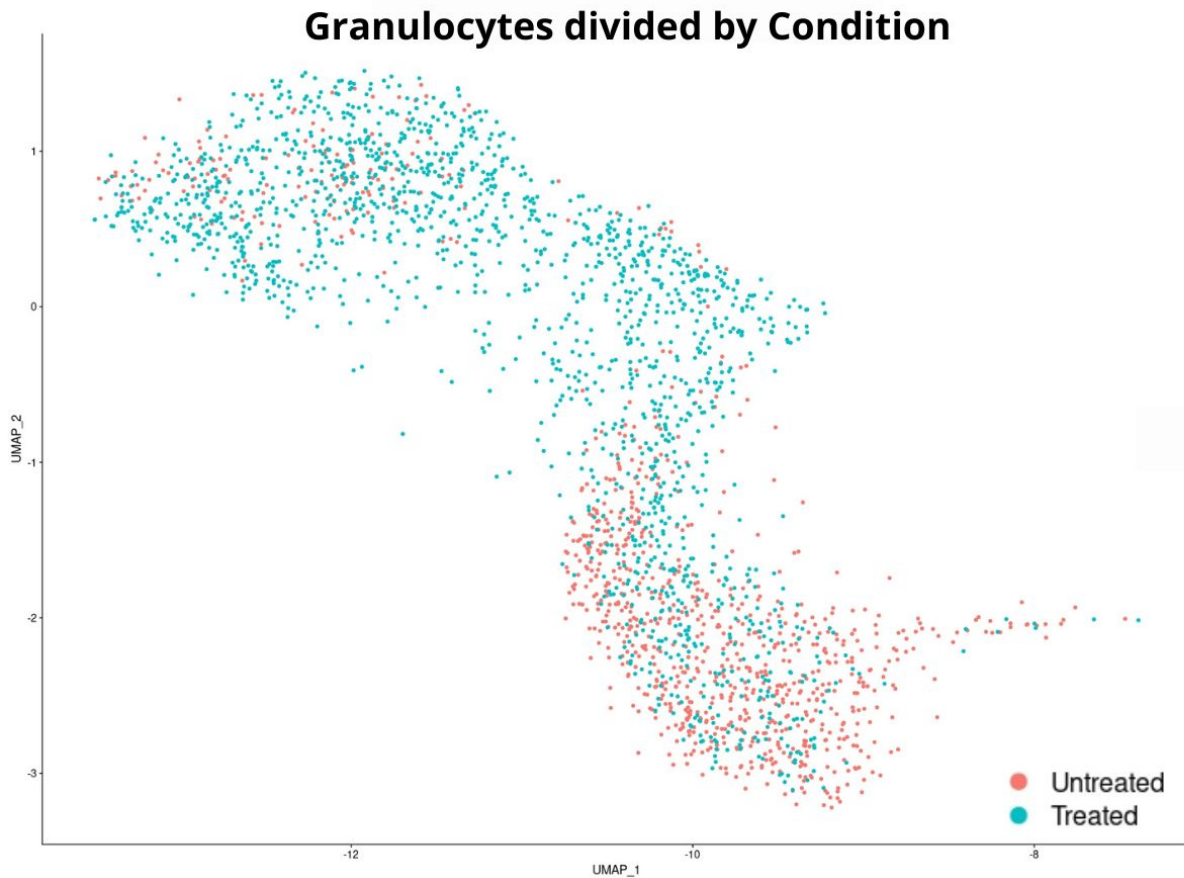


Figure B.31: Granulocytes divided by Condition

Código B.30: R studio code extract – Manual selection: Dendritic cells

```

1 #- Sell selection -#
2 select.cells <- CellSelector(plot = plot)
3 Idents(samplesIntegrated, cells = select.cells) <- "Dendritic ←
  ↪ cells"
4 plot <- DimPlot(samplesIntegrated, reduction = "umap", label = ←
  ↪ TRUE, pt.size = 1)
5
6 #- Subset -#
7 samplesIntegrated_DC <- subset(samplesIntegrated, idents = "←
  ↪ Dendritic cells")
8
9 DimPlot(samplesIntegrated_DC, reduction = "umap", pt.size = 1.5, ←
  ↪ cols = "lightseagreen", group.by = "integrated_snn_res.0") +←
  ↪ ggtitle('Dendritic cells')
10
11 DimPlot(samplesIntegrated_DC, reduction = "umap", label = FALSE, ←
  ↪ pt.size = 1.5, group.by = 'Treatment') + ggtitle('Dendritic ←
  ↪ cells by Treatment')
```

12

```
13 #- Save data -#
```

```
14 saveRDS(samplesIntegrated_DC, "samplesIntegrated_DCSelect.rds")
```

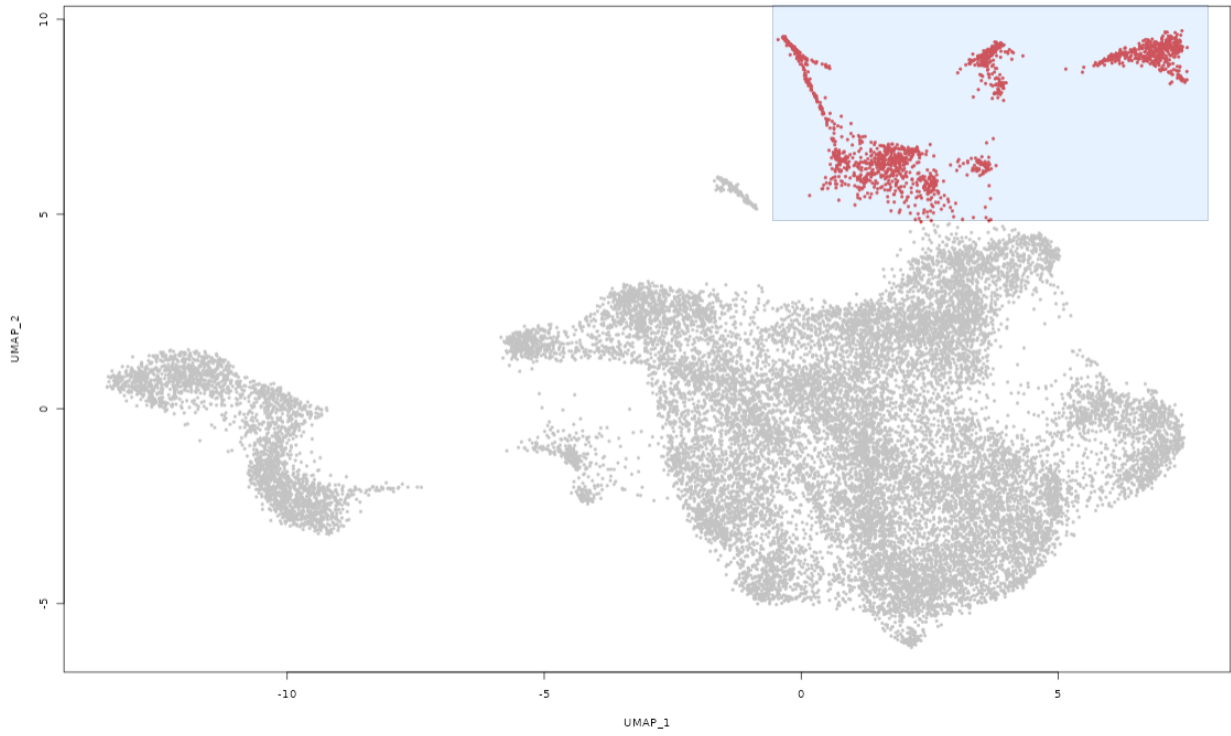


Figure B.32: Dendritic cells manual selection

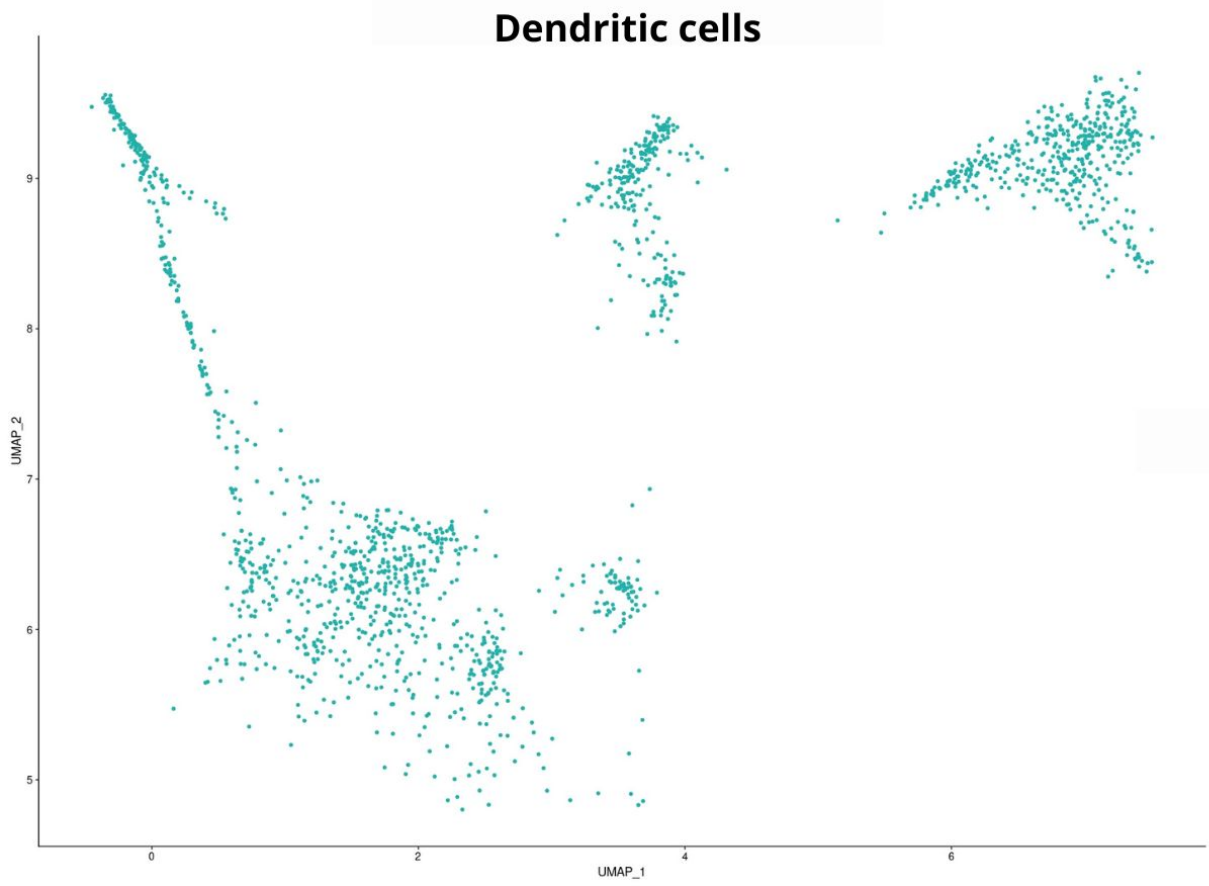


Figure B.33: Dendritic cells

Dendritic cells divided by Condition

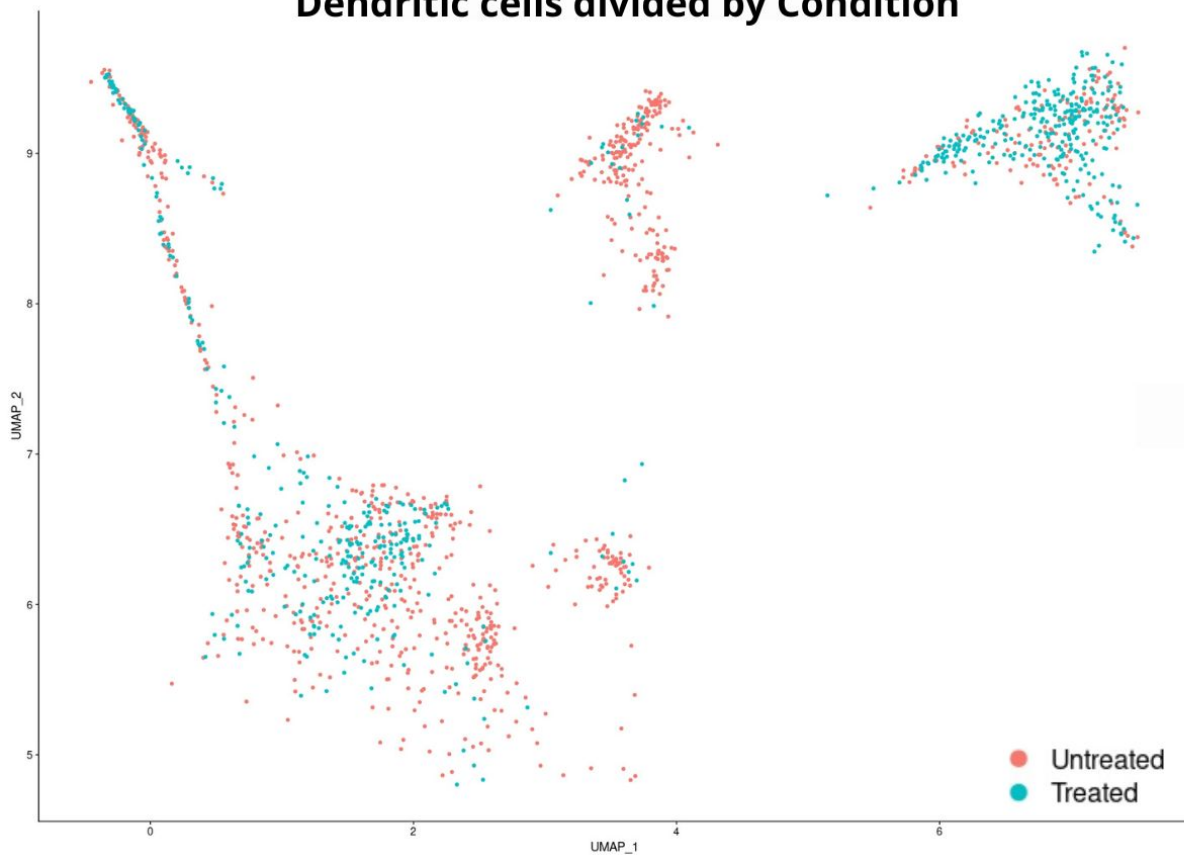


Figure B.34: Dendritic cells divided by Condition

Código B.31: R studio code extract – Manual selection: Monocyte/-
Macrophages

```
1 #- Sell selection -#
2 select.cells <- CellSelector(plot = plot)
3 Idents(samplesIntegrated, cells = select.cells) <- "Monocyte/↔
  ↳ Macrophages"
4
5 select.cells <- CellSelector(plot = plot)
6 Idents(samplesIntegrated, cells = select.cells) <- "Monocyte/↔
  ↳ Macrophages"
7
8 plot <- DimPlot(samplesIntegrated, reduction = "umap", label = ↔
  ↳ TRUE, pt.size = 1)
9
10 #- Subset -#
11 samplesIntegrated_MoMa <- subset(samplesIntegrated, idents = "↔
  ↳ Monocyte/Macrophages")
12
13 DimPlot(samplesIntegrated_MoMa, reduction = "umap", pt.size = 1.5, ↔
```

```

14     ↪ cols = "darkviolet", group.by = "integrated_snn_res.0") + ↪
15     ↪ ggtitle('Monocyte/Macrophages')
15 DimPlot(samplesIntegrated_MoMa, reduction = "umap", label = FALSE, ↪
16     ↪ pt.size = 1.5, group.by = 'Treatment') + ggtitle('Monocyte/ ↪
17     ↪ Macrophages by Treatment')
16
17 #- Save data -#
18 saveRDS(samplesIntegrated_MoMa, "samplesIntegrated_MoMaSelect.rds")

```

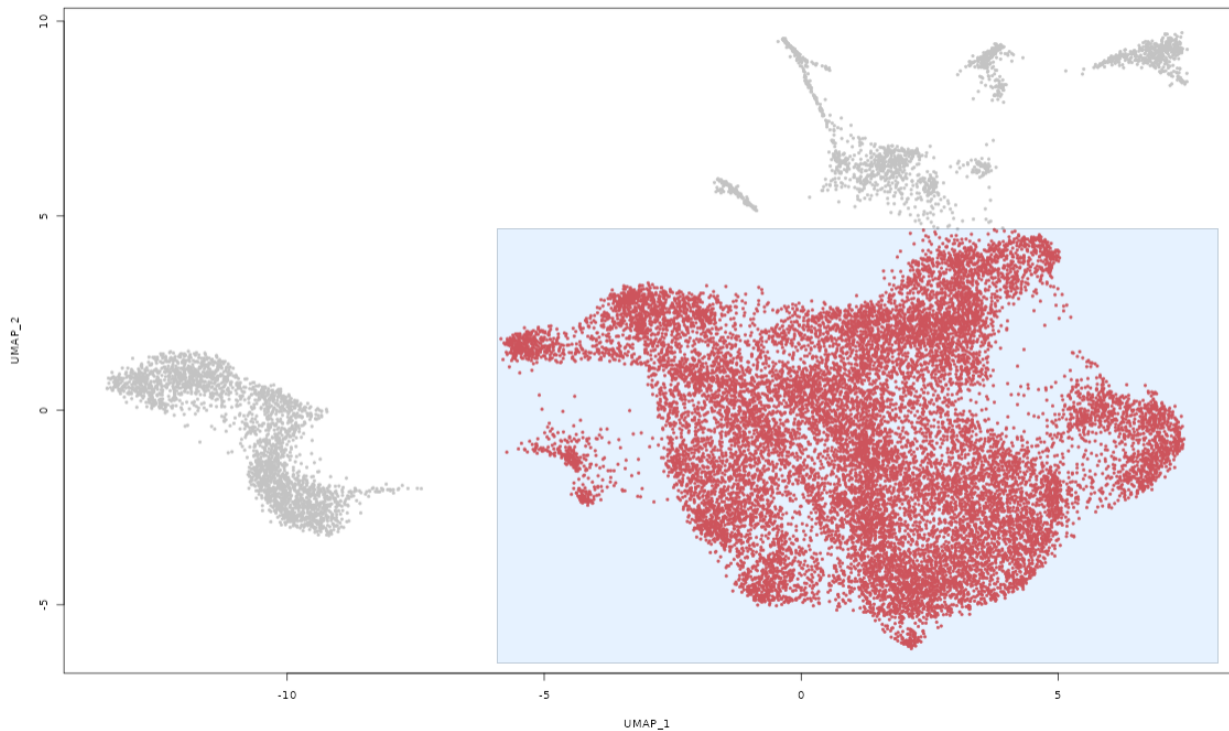


Figure B.35: Monocyte/Macrophages manual selection part 1

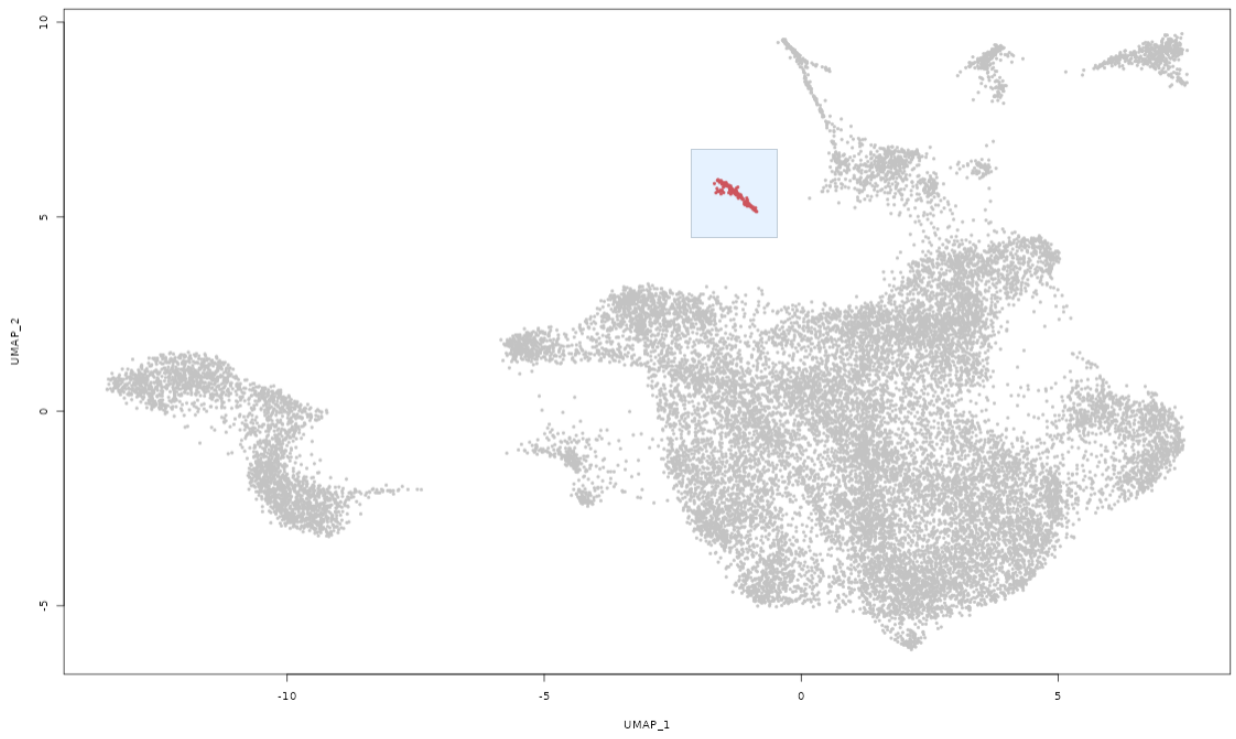


Figure B.36: Monocyte/Macrophages manual selection part 2

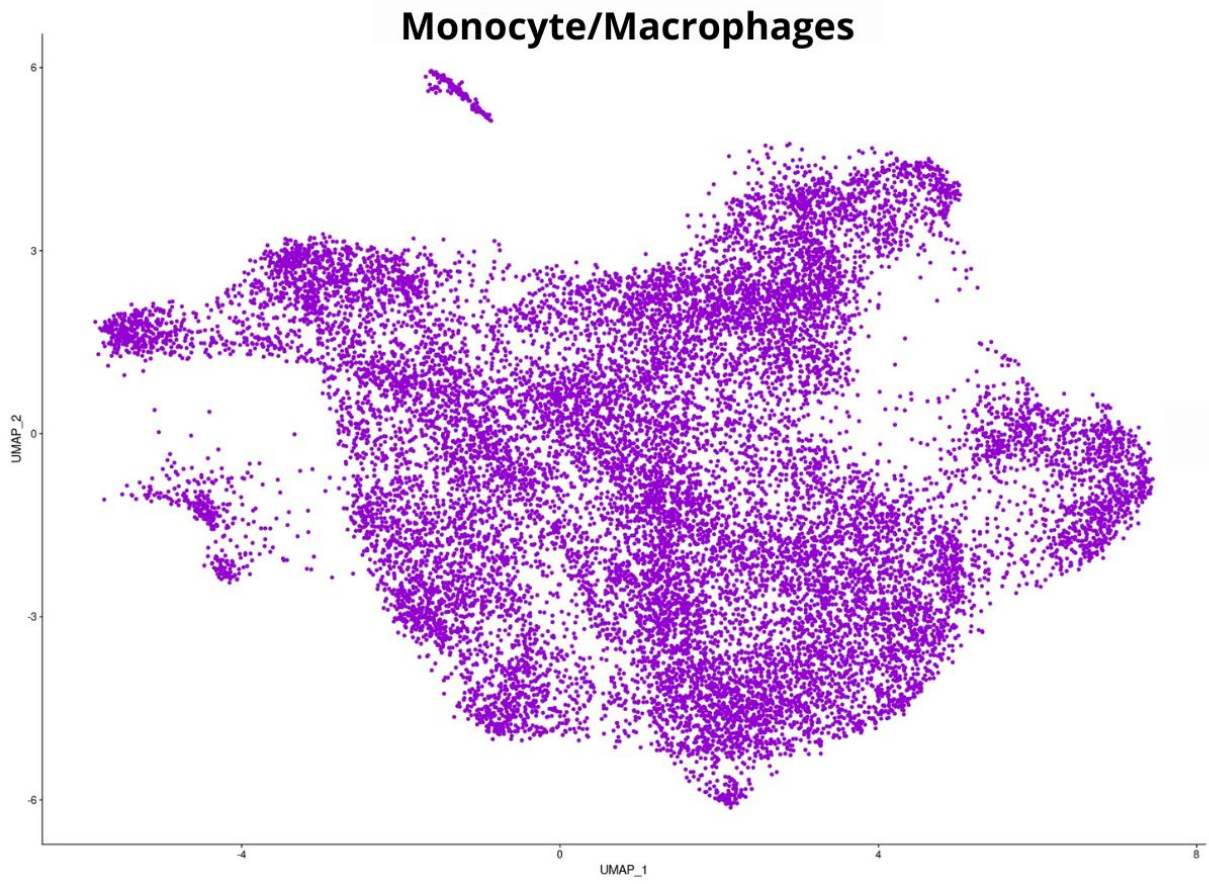


Figure B.37: Monocyte/Macrophages

Monocyte/Macrophages divided by Condition

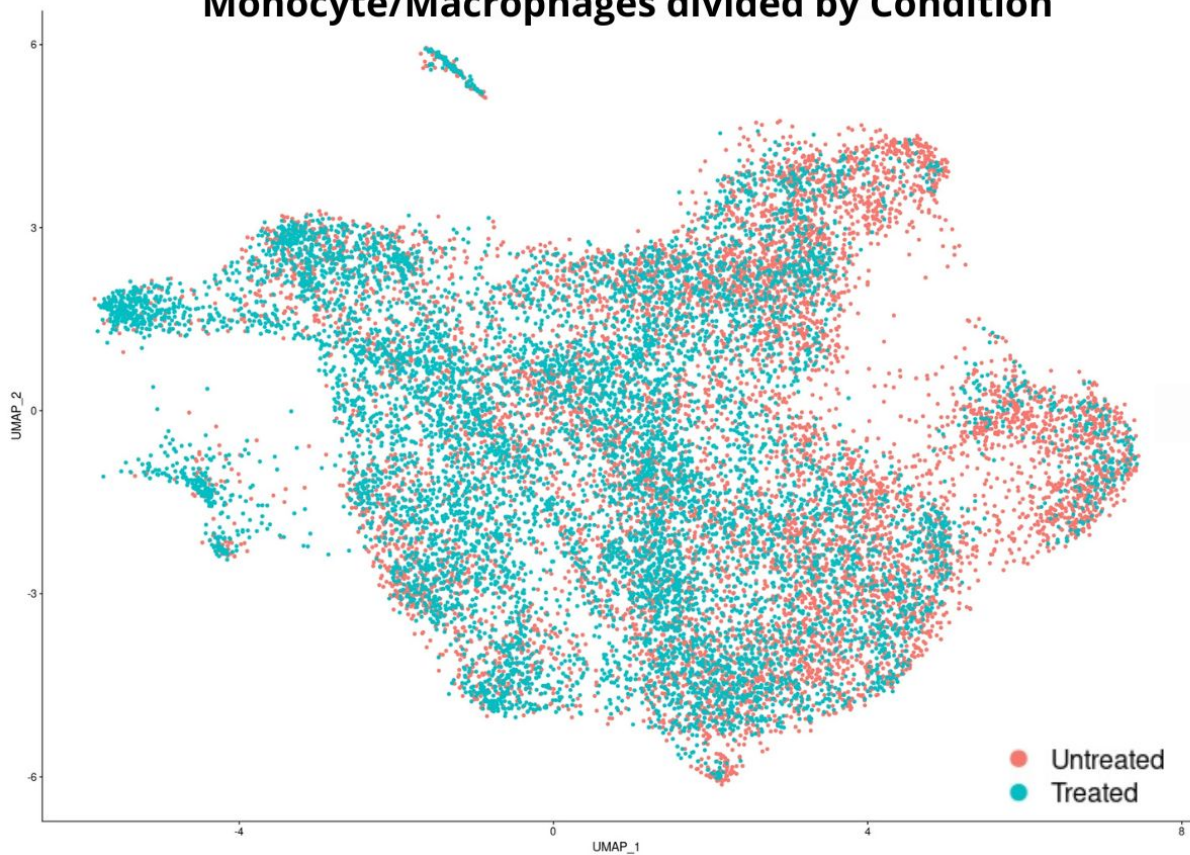


Figure B.38: Monocyte/Macrophages divided by Condition

Código B.32: R studio code extract – Population assignment

```
1
2 #- Assing population identity -#
3 samplesIntegrated[["Population"]] <- samplesIntegrated@active.<-
  <- ident
4
5 #- Final UMAP -#
6
7 DimPlot(samplesIntegrated, reduction = "umap", label = TRUE, pt.<-
  <- size = 1.5, cols = my_colors, group.by = "Population")
```

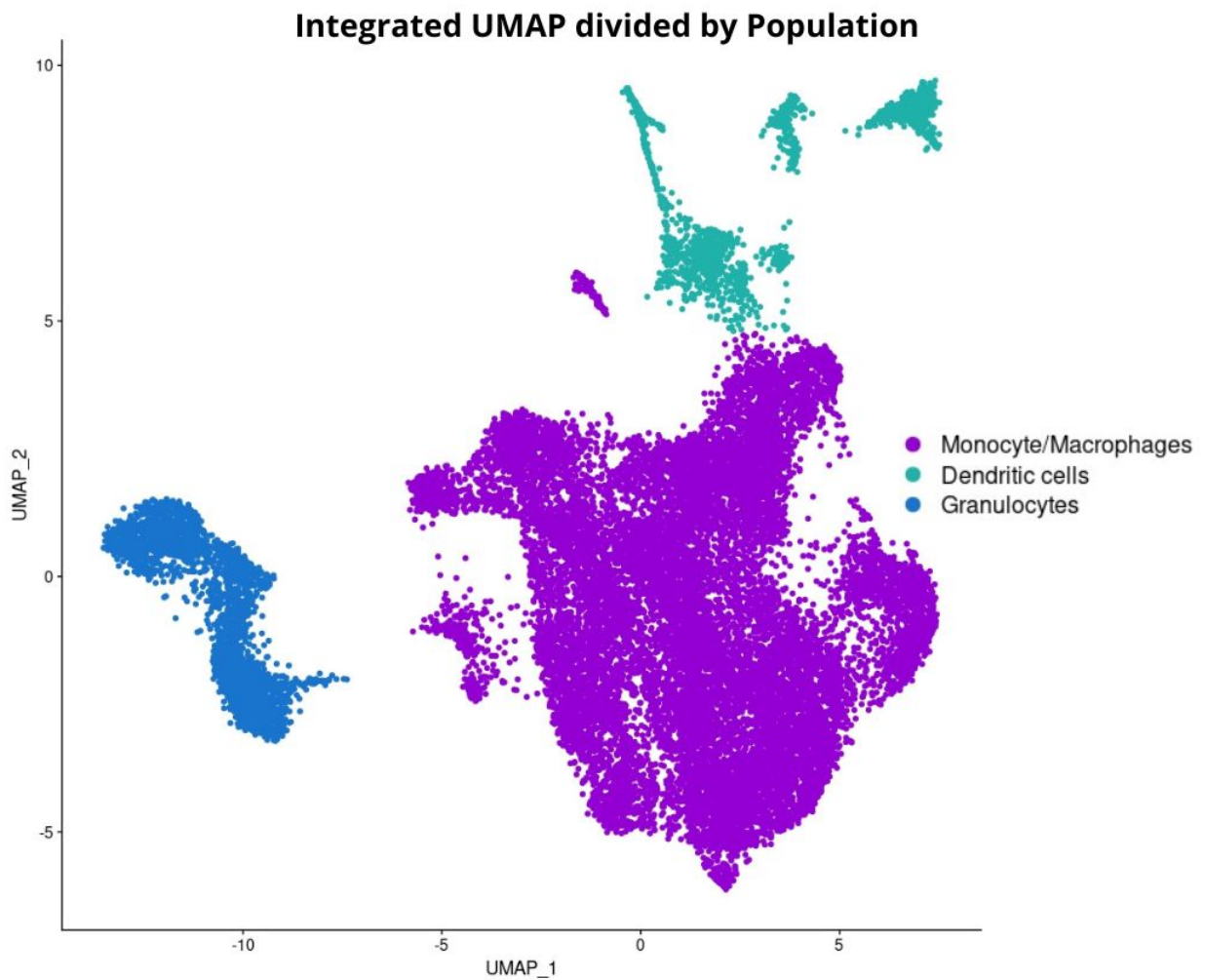



Figure B.39: Classified populations UMAP

Código B.33: R studio code extract – Density plots

```

1 #--- Density plot ---#
2 DefaultAssay(object = samplesIntegrated) <- "SCT"
3
4 sampList <- SplitObject(samplesIntegrated, split.by = "Treatment" ↵
  ↵ )
5
6 Sample_Untreated <- sampList[[1]]
7 Sample_Treated <- sampList[[2]]
8
9 saveRDS(Sample_Untreated, "Sample_Untreated.rds")
10 saveRDS(Sample_Treated, "Sample_Treated.rds")
11
12 #- Normal -#

```

```

13 p <- DimPlot(samplesIntegrated, group.by = "Treatment", reduction =<-
    ↪ "umap")
14
15 #Prepare data
16 data <- data.frame(samplesIntegrated@meta.data)
17 data$UMAP1 <- Embeddings(samplesIntegrated, "umap")[,1]
18 data$UMAP2 <- Embeddings(samplesIntegrated, "umap")[,2]
19
20 #Create a data frame that doesn't contain a "sample" column.
21 data_bg <- data[, -(which(colnames(data))=="Sample")]
22
23 #Plot
24 density_plot <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
25   geom_point(colour="#00000000") +
26   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins<-
    ↪ =25) +
27   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
28   xlim(c(-15, 10)) + ylim(c(-10, 12)) +
29   #facet_wrap(~Treatment, ncol=3)+
30   theme_classic() + ggtitle("Density plot") +
31   theme(plot.title = element_text(hjust = 0.5), text = element_↪
    ↪ text(size = 14))
32
33 density_plot
34
35 #- density 2: Untreated -#
36
37 #Prepare data
38 data <- data.frame(Sample_Untreated@meta.data)
39 data$UMAP1 <- Embeddings(Sample_Untreated, "umap")[,1]
40 data$UMAP2 <- Embeddings(Sample_Untreated, "umap")[,2]
41
42 #Create a data frame that doesn't contain a "sample" column.
43 data_bg <- data[, -(which(colnames(data))=="Sample")]
44
45 #Plot
46 density_plot_2 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
47   geom_point(colour="#00000000") +
48   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins<-
    ↪ =25) +
49   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
50   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
51   facet_wrap(~Treatment, ncol=3)+

```



```

52 theme_classic() + ggtitle("Untreated Density plot") +
53 theme(plot.title = element_text(hjust = 0.5), text = element_↵
↵ text(size = 14))
54
55 density_plot_2
56
57 #- density 3: Untreated -#
58
59 #Prepare data
60 data <- data.frame(Sample_Treated@meta.data)
61 data$UMAP1 <- Embeddings(Sample_Treated, "umap")[,1]
62 data$UMAP2 <- Embeddings(Sample_Treated, "umap")[,2]
63
64 #Create a data frame that doesn't contain a "sample" column.
65 data_bg <- data[, -(which(colnames(data)=="Sample"))]
66
67 #Plot
68 density_plot_3 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
69   geom_point(colour="#00000000") +
70   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↵
↵ =25) +
71   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↵
↵ darkolivegreen3", "goldenrod1", "red")) +
72   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
73   facet_wrap(~Treatment, ncol=3)+
74   theme_classic() + ggtitle("Treated Density plot") +
75   theme(plot.title = element_text(hjust = 0.5), text = element_↵
↵ text(size = 14))
76
77 density_plot_3
78
79
80 plot_grid(p, density_plot, density_plot_2, density_plot_3, ncol=2)

```

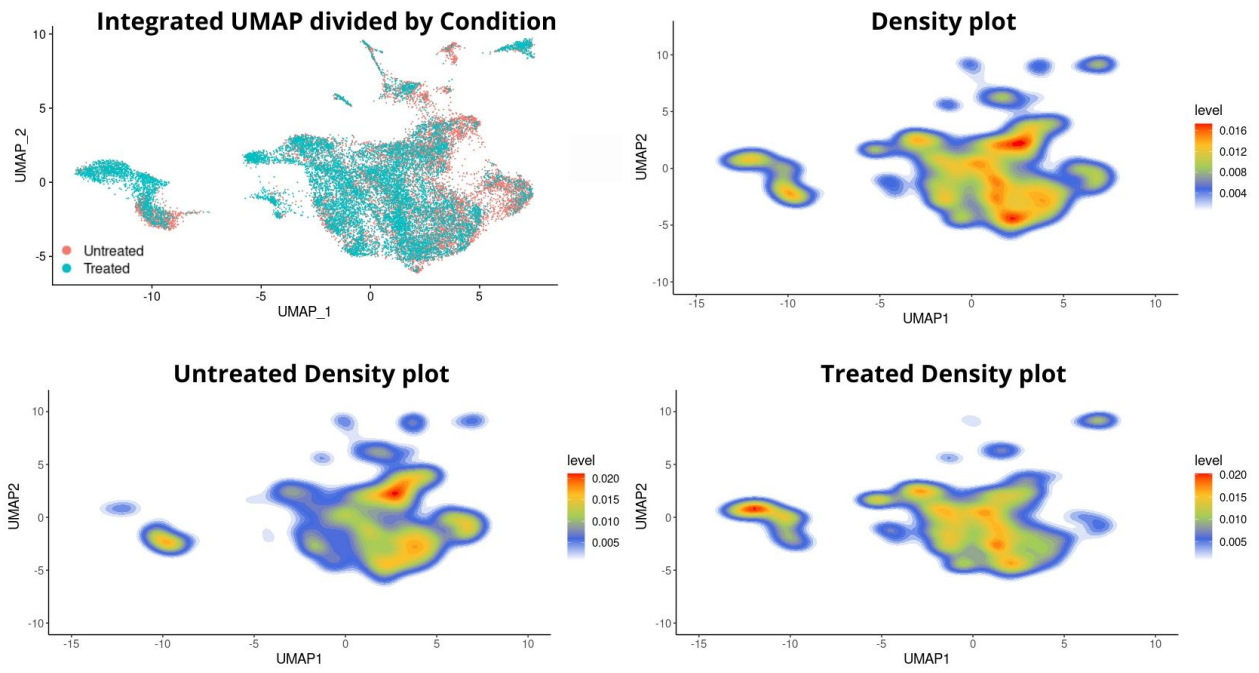


Figure B.40: Density plots

Código B.34: R studio code extract – Myeloid data Barplot

```

1
2 samplesIntegrated[["Subpopulation"]] <- samplesIntegrated@active.<-
  <- ident
3 dittoBarPlot(object = samplesIntegrated, scale = "percent", var = <-
  <- "Subpopulation", group.by = "Treatment", color.panel = my_<-
  <- colors, x.labels = c("Untreated", "Treated"), xlab = "<-
  <- Condition", main = "")

```

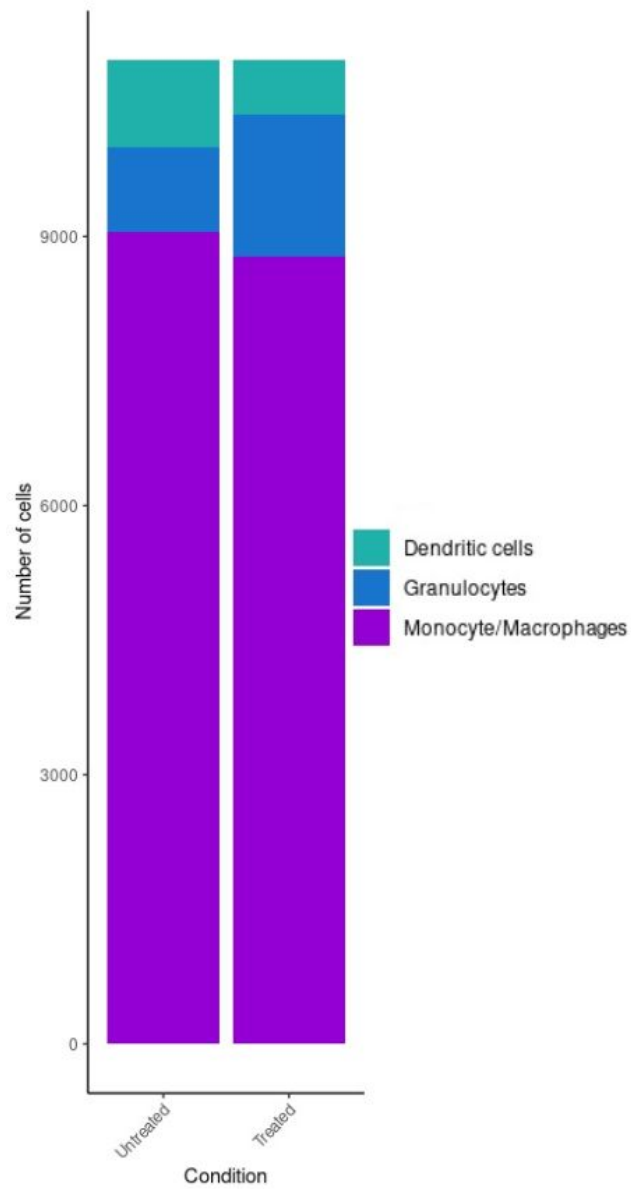


Figure B.41: Barplots of myeloid cells populations number by Condition

Código B.35: R studio code extract – Myeloid data populations divided by Sample

```
1 DimPlot(samplesIntegrated, reduction = "umap", split.by = 'orig.↵
  ↵ ident', pt.size = 1.5, ncol = 2, cols = my_colors)
```

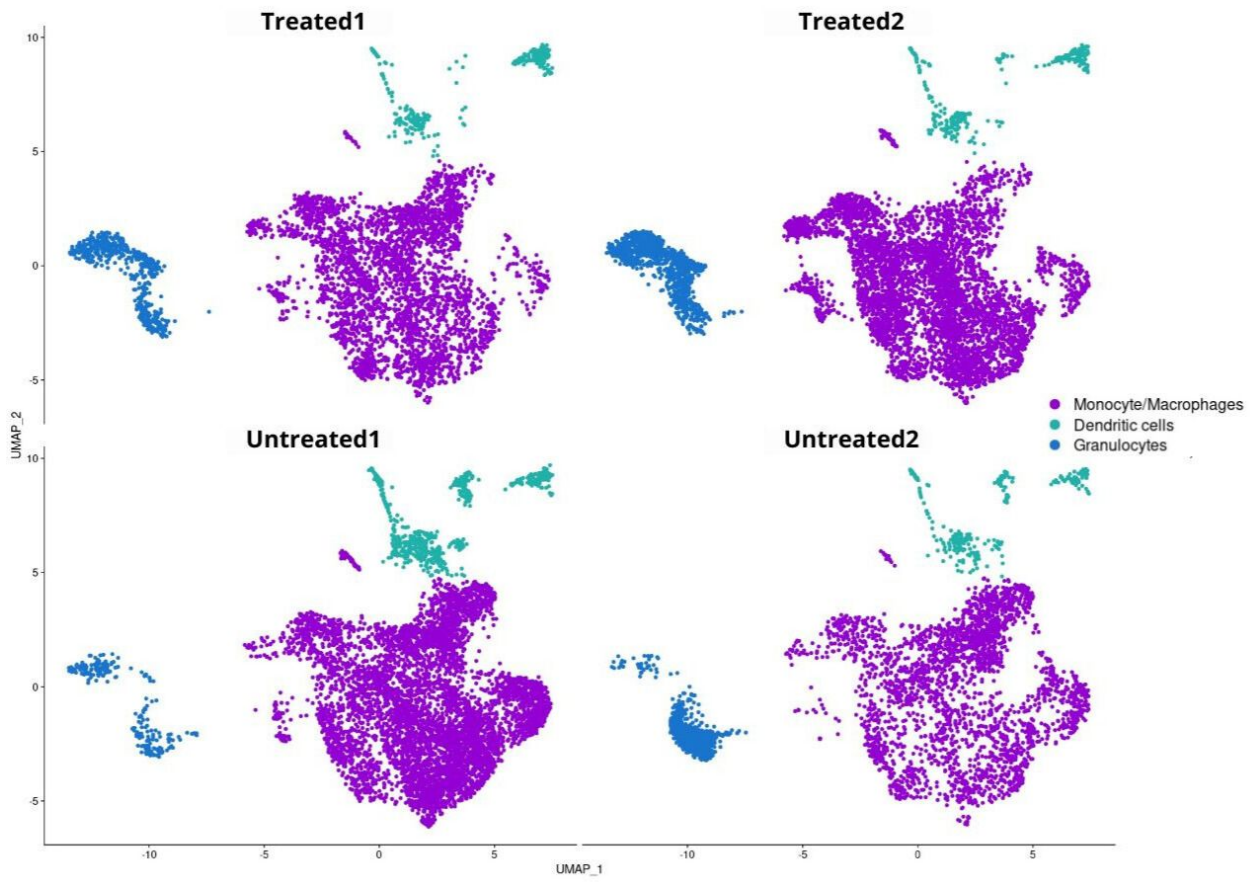


Figure B.42: UMAPs of myeloid cells populations divided by Sample

Annex C. Dendritic cells mice analysis code

Código C.1: R studio code extract – Quality Control Dendritic cells Part 1

```
1 #----- Dendritic cells analysis -----#
2
3 #- Separate cell by sample -#
4 samplesIntegrated_DC.list <- SplitObject(samplesIntegrated_DC, ↵
  ↵ split.by = "orig.ident")
5 Untreated1 <- samplesIntegrated_DC.list [[1]]
6 Untreated2 <- samplesIntegrated_DC.list [[2]]
7 Treated1 <- samplesIntegrated_DC.list [[3]]
8 Treated2 <- samplesIntegrated_DC.list [[4]]
9
10 #- Check nFeature distribution with Histograms -#
11 Untreated1 <- FetchData(Untreated1, vars=c("nFeature_RNA"))
12 Untreated1 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins↵
  ↵ =100) + geom_vline(xintercept = 2000) + ggtitle('Untreated1'↵
  ↵ )
13
14 Untreated2 <- FetchData(Untreated2, vars=c("nFeature_RNA"))
15 Untreated2 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins↵
  ↵ =100) + geom_vline(xintercept = 2000) + ggtitle('Untreated2'↵
  ↵ )
16
17 Treated1 <- FetchData(Treated1, vars=c("nFeature_RNA"))
18 Treated1 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins↵
  ↵ =100) + geom_vline(xintercept = 2000) + ggtitle('Treated1')
19
20 Treated2 <- FetchData(Treated2, vars=c("nFeature_RNA"))
21 Treated2 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins↵
  ↵ =100) + geom_vline(xintercept = 2000) + ggtitle('Treated2')
```

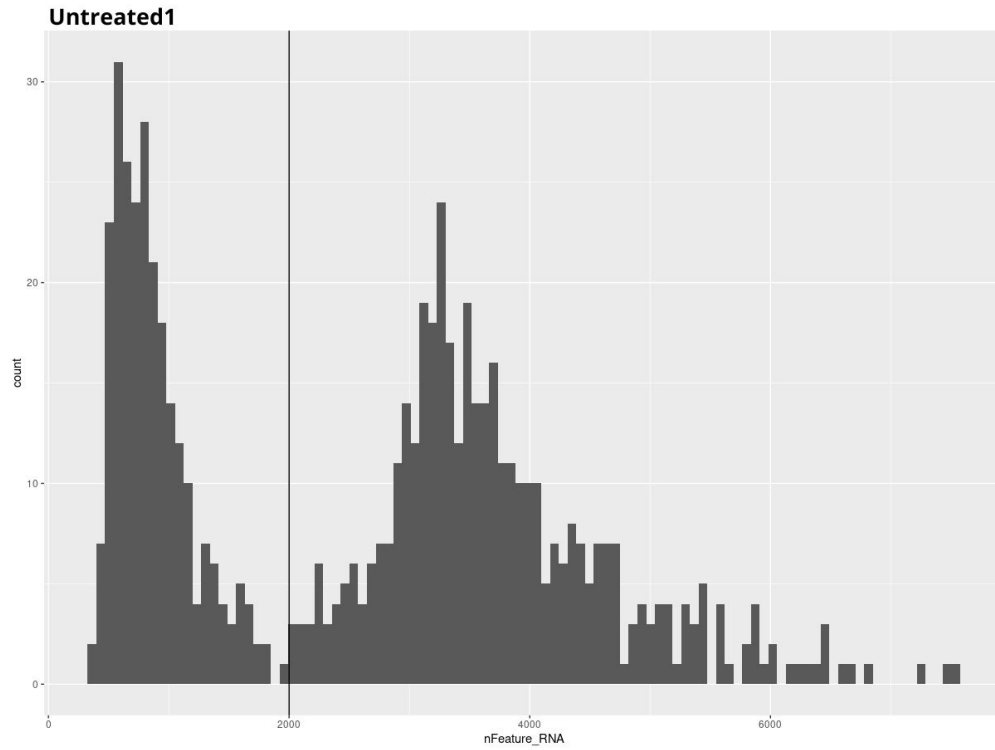


Figure C.1: Histogram of the number of features expressed vs number of cells for Dendritic cells Untreated sample 1

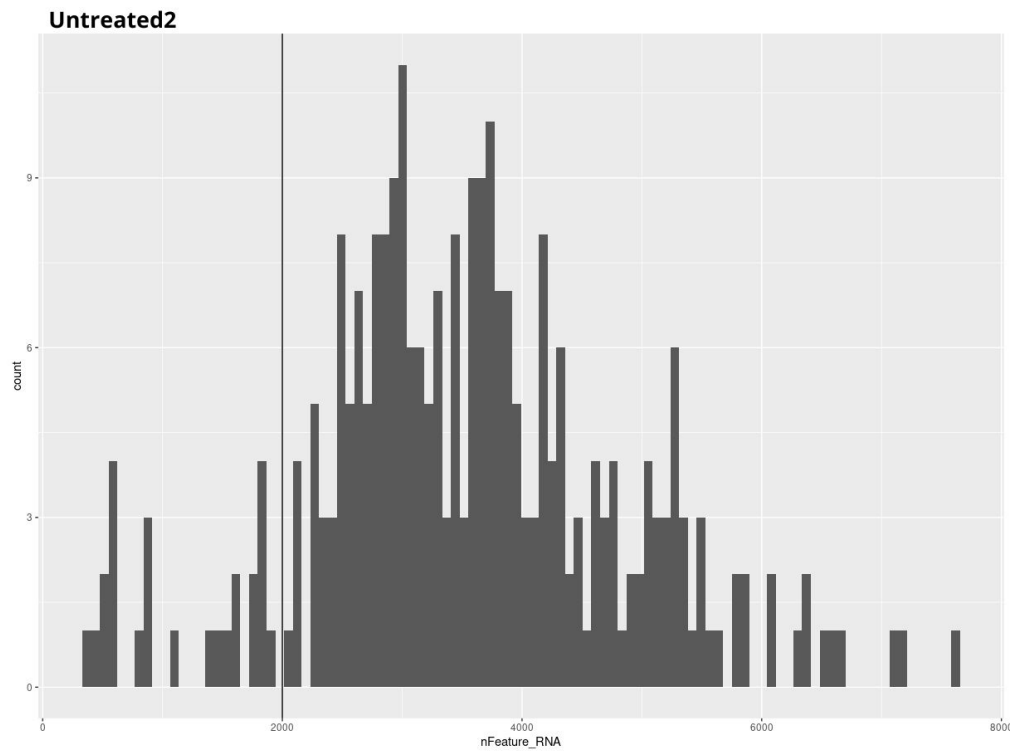


Figure C.2: Histogram of the number of features expressed vs number of cells for Dendritic cells Untreated sample 2

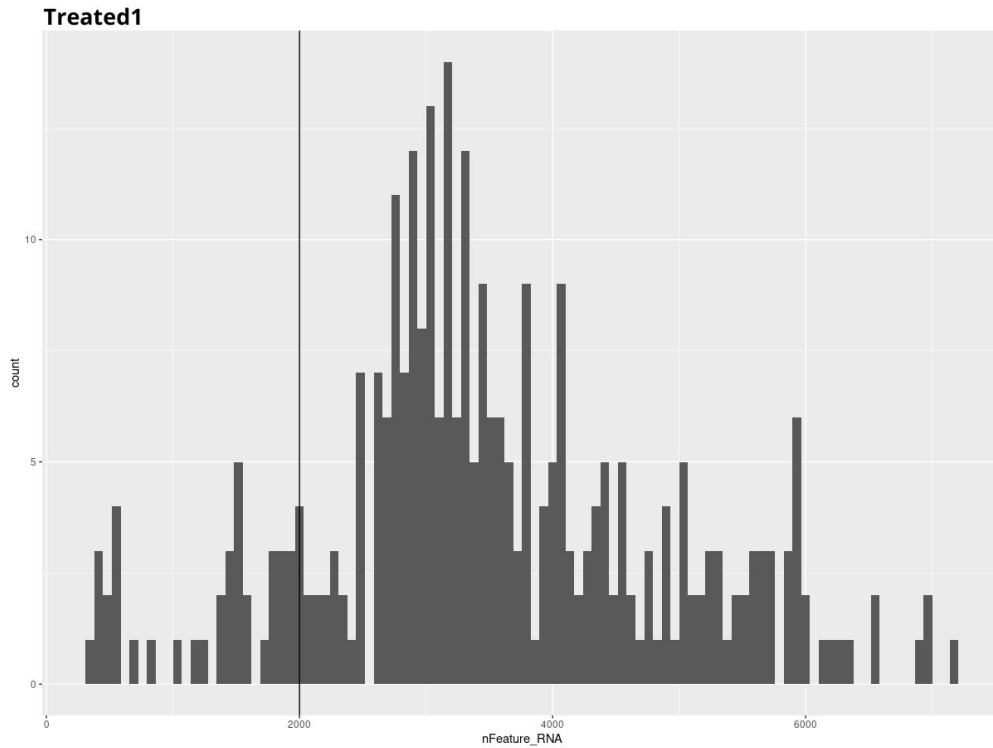


Figure C.3: Histogram of the number of features expressed vs number of cells for Dendritic cells Treated sample 1

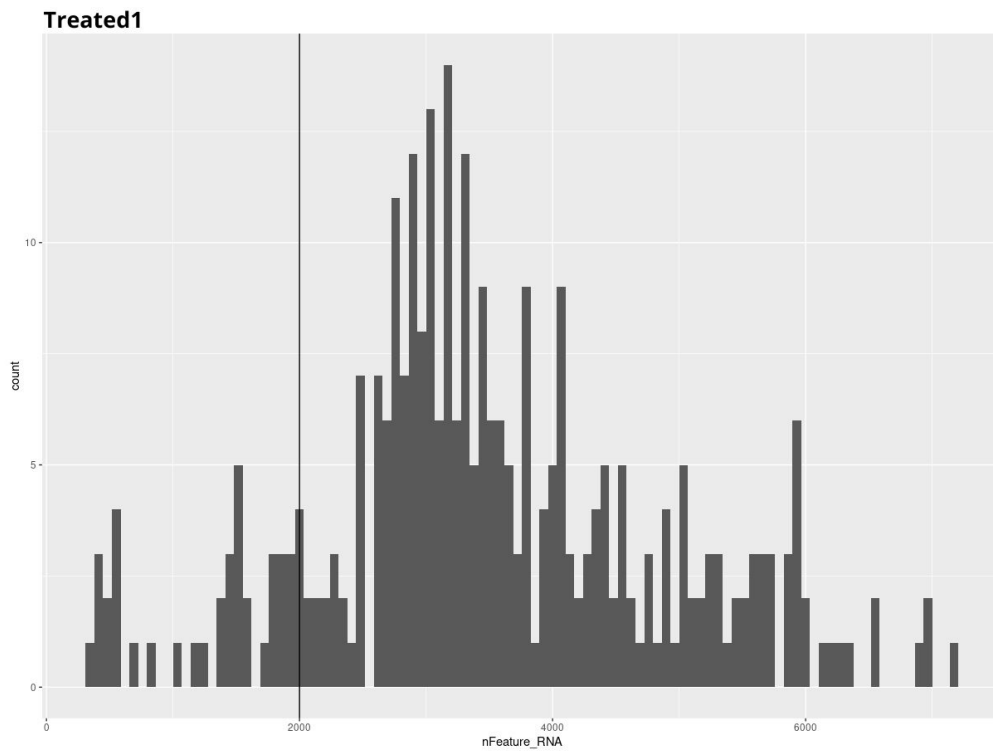


Figure C.4: Histogram of the number of features expressed vs number of cells for Dendritic cells Treated sample 2

Código C.2: R studio code extract – Quality Control Dendritic cells Part 2

```

1  #- Quality Control with Scatter plots -#
2  Untreated1 <- samplesIntegrated_DC.list [[1]]
3  Untreated2 <- samplesIntegrated_DC.list [[2]]
4  Treated1 <- samplesIntegrated_DC.list [[3]]
5  Treated2 <- samplesIntegrated_DC.list [[4]]
6
7  #- Untreated 1 -#
8  #Logarithmic data
9  logMT <- FetchData(Untreated1, vars=c("percent.mt"))
10 nFeature <- FetchData(Untreated1, vars=c("nFeature_RNA"))
11
12 Untreated1[["logMT"]] <- log10(logMT)
13 Untreated1[["logFeature"]] <- log10(nFeature)
14
15 #Scatter plots
16 p1 <- FeatureScatter(Untreated1, feature1="nFeature_RNA", feature2=<-
  <- "percent.mt", group.by = "orig.ident") + geom_hline(<-
  <- yintercept=10) + ggtitle('Untreated1') + geom_vline(<-
  <- xintercept = 2000)
17 p2 <- FeatureScatter(Untreated1, feature1="logFeature", feature2="<-
  <- logMT", group.by = "orig.ident") + geom_hline(yintercept=1) <-
  <- + ggtitle('Untreated1') + geom_vline(xintercept = 3.301)
18 p1+p2
19
20 #Create subset with new data
21 Untreated1 <- subset(Untreated1, subset = nFeature_RNA > 2000)
22
23
24 #- Untreated 2 -#
25 #Logarithmic data
26 logMT <- FetchData(Untreated2, vars=c("percent.mt"))
27 nFeature <- FetchData(Untreated2, vars=c("nFeature_RNA"))
28
29 Untreated2[["logMT"]] <- log10(logMT)
30 Untreated2[["logFeature"]] <- log10(nFeature)
31
32 #Initial Scatter plots
33 p1 <- FeatureScatter(Untreated2, feature1="nFeature_RNA", feature2=<-
  <- "percent.mt", group.by = "orig.ident") + geom_hline(<-
  <- yintercept=10) + ggtitle('Untreated2') + geom_vline(<-
  <- xintercept = 2000)
34 p2 <- FeatureScatter(Untreated2, feature1="logFeature", feature2="<-
  <- logMT", group.by = "orig.ident")+ geom_hline(yintercept=1) +<-
  <- ggtitle('Untreated2') + geom_vline(xintercept = 3.301)

```



```

35 p1+p2
36
37 #Create subset with new data
38 Untreated2 <- subset(Untreated2, subset = nFeature_RNA > 2000)
39
40
41 #- Treated 1 -#
42 #Logarithmic data
43 logMT <- FetchData(Treated1, vars=c("percent_mt"))
44 nFeature <- FetchData(Treated1, vars=c("nFeature_RNA"))
45
46 Treated1[["logMT"]] <- log10(logMT)
47 Treated1[["logFeature"]] <- log10(nFeature)
48
49 #Scatter plots
50 p1 <- FeatureScatter(Treated1, feature1="nFeature_RNA", feature2="↵
↵ percent_mt", group.by = "orig.ident") + geom_hline(↵
↵ yintercept=7) + ggtitle('Treated1') + geom_vline(xintercept ↵
↵ = 2000)
51 p2 <- FeatureScatter(Treated1, feature1="logFeature", feature2="↵
↵ logMT", group.by = "orig.ident") + geom_hline(yintercept↵
↵ =0.85) + ggtitle('Treated1') + geom_vline(xintercept = ↵
↵ 3.301)
52 p1+p2
53
54 #Create subset with new data
55 Treated1 <- subset(Treated1, subset = nFeature_RNA > 2000)
56
57
58 #- Treated 2 -#
59 #Log graphics
60 logMT <- FetchData(Treated2, vars=c("percent_mt"))
61 nFeature <- FetchData(Treated2, vars=c("nFeature_RNA"))
62
63 Treated2[["logMT"]] <- log10(logMT)
64 Treated2[["logFeature"]] <- log10(nFeature)
65
66 #Scatter plots
67 p1 <- FeatureScatter(Treated2, feature1="nFeature_RNA", feature2="↵
↵ percent_mt", group.by = "orig.ident") + geom_hline(↵
↵ yintercept=8) + ggtitle('Treated2') + geom_vline(xintercept ↵
↵ = 2000)
68 p2 <- FeatureScatter(Treated2, feature1="logFeature", feature2="↵
↵ logMT", group.by = "orig.ident") + geom_hline(yintercept↵
↵ =0.9) + ggtitle('Treated2') + geom_vline(xintercept = 3.301)

```

```
69 p1+p2
70
71 #Create subset with new data
72 Treated2 <- subset(Treated2, subset = nFeature_RNA > 2000)
```

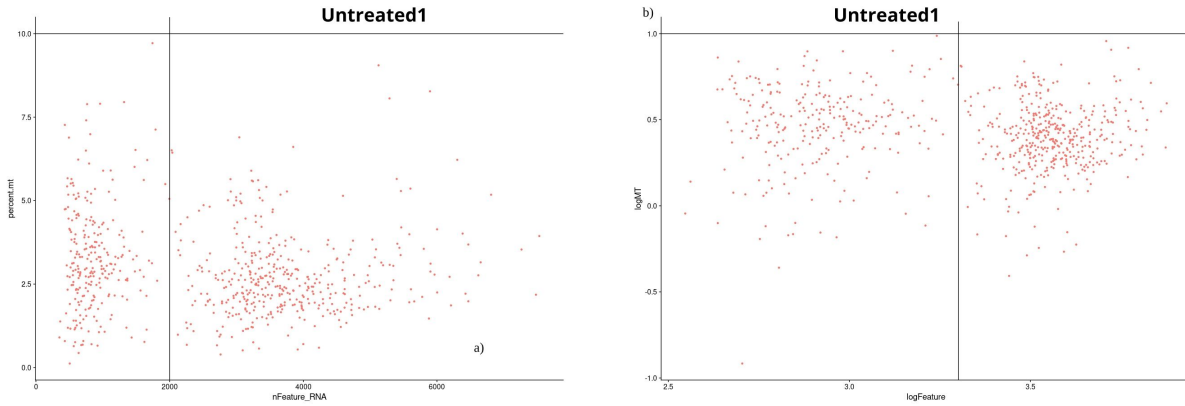


Figure C.5: Scatter plot of Dendritic cells Untreated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

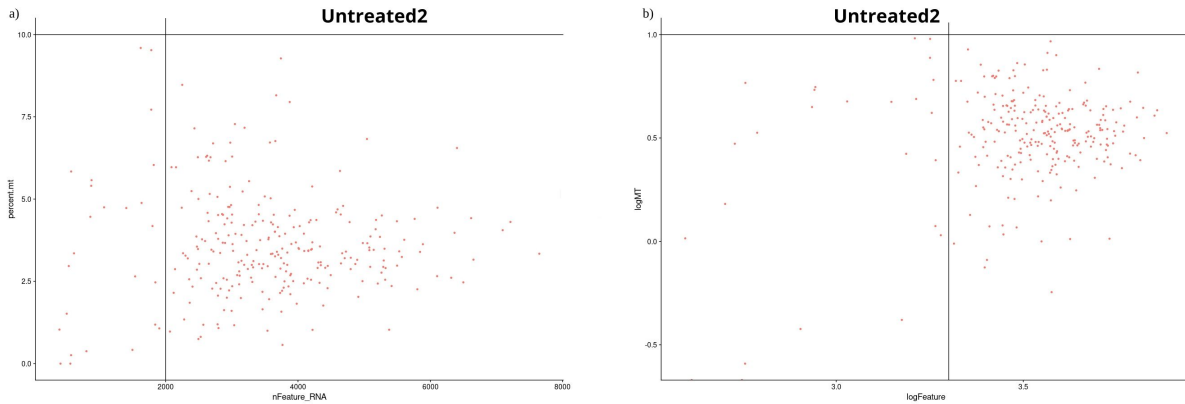


Figure C.6: Scatter plot of Dendritic cells Untreated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

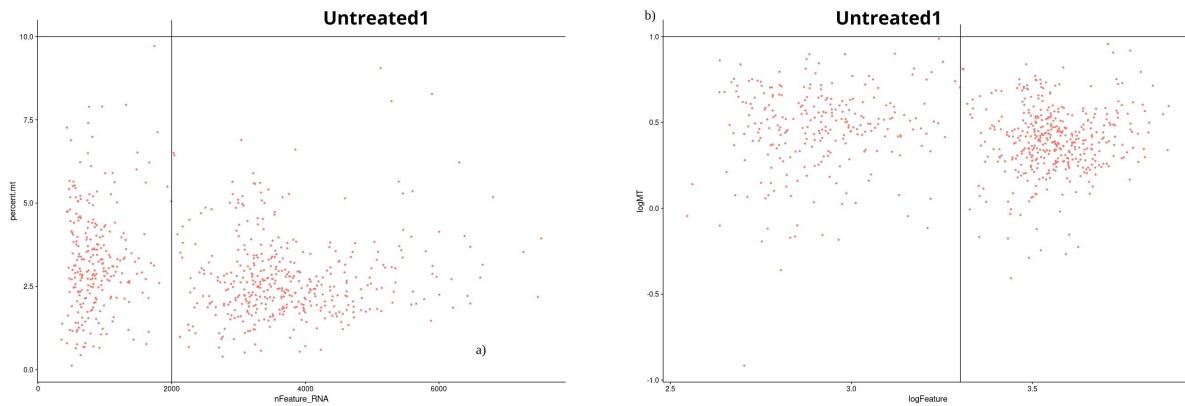


Figure C.7: Scatter plot of Dendritic cells Treated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

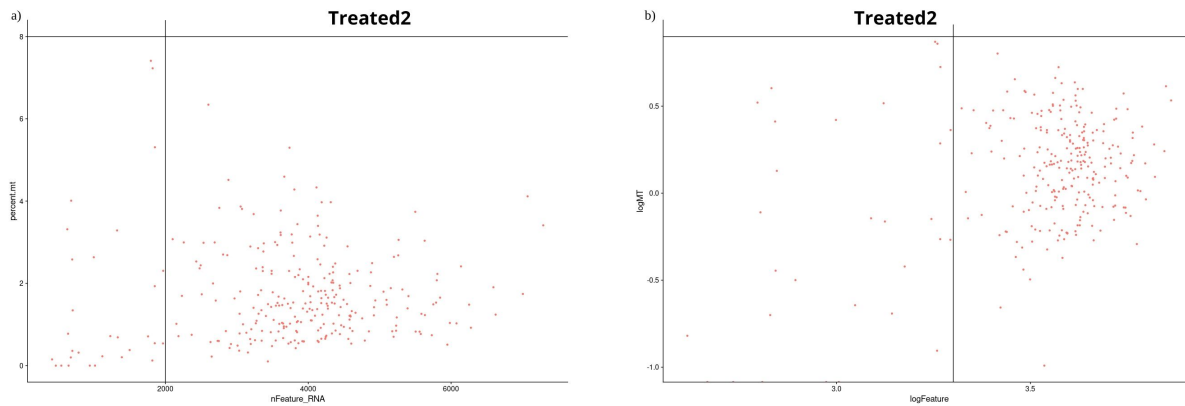


Figure C.8: Scatter plot of Dendritic cells Treated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

Código C.3: R studio code extract – Dendritic cell Integration

```

1 #--- Integration ---#
2
3 samplesDC.list <- list(Untreated1, Untreated2, Treated1, Treated2)
4
5 #- Normalice samples -#
6 samplesDC.list <- lapply(X= samplesDC.list, FUN = function(x){
7   x <- SCTransform(object = x, vars.to.regress = "percent.mt",
8     verbose = FALSE, variable.features.n = 5000)})
9
10 #- Select integration features and prepare for integration -#
11 samplesDC.features <- SelectIntegrationFeatures(object.list =
12   samplesDC.list )

```

```

11 samplesDC.list <- PrepSCTIntegration(object.list = samplesDC.list , ←
    ↪ anchor.features = samplesDC.features)
12
13 #- Select integration anchors -#
14 samplesDC.anchors <- FindIntegrationAnchors(object.list = ←
    ↪ samplesDC.list , normalization.method = "SCT" , anchor.←
    ↪ features = samplesDC.features)
15 save(samplesDC.anchors , file=file.path("~/", "samplesDC.anchors.←
    ↪ rdata"))
16
17 #- Integrate -#
18 samplesIntegFltDC <- IntegrateData(anchorset = samplesDC.anchors , ←
    ↪ normalization.method = "SCT")
19 saveRDS(samplesIntegFltDC , "samplesIntegFltDC.rds")

```

Código C.4: R studio code extract – Dendritic cell linear reduction

```

1 #- PCA reduction -#
2 samplesIntegFltDC <- RunPCA(samplesIntegFltDC , npcs = 50 , verbose=F←
    ↪ )
3
4 #Elbow plot
5 ElbowPlot(samplesIntegFltDC , ndims = 50 , reduction = "pca") + ←
    ↪ ggtitle("Elbow plot Integrated DC data")

```

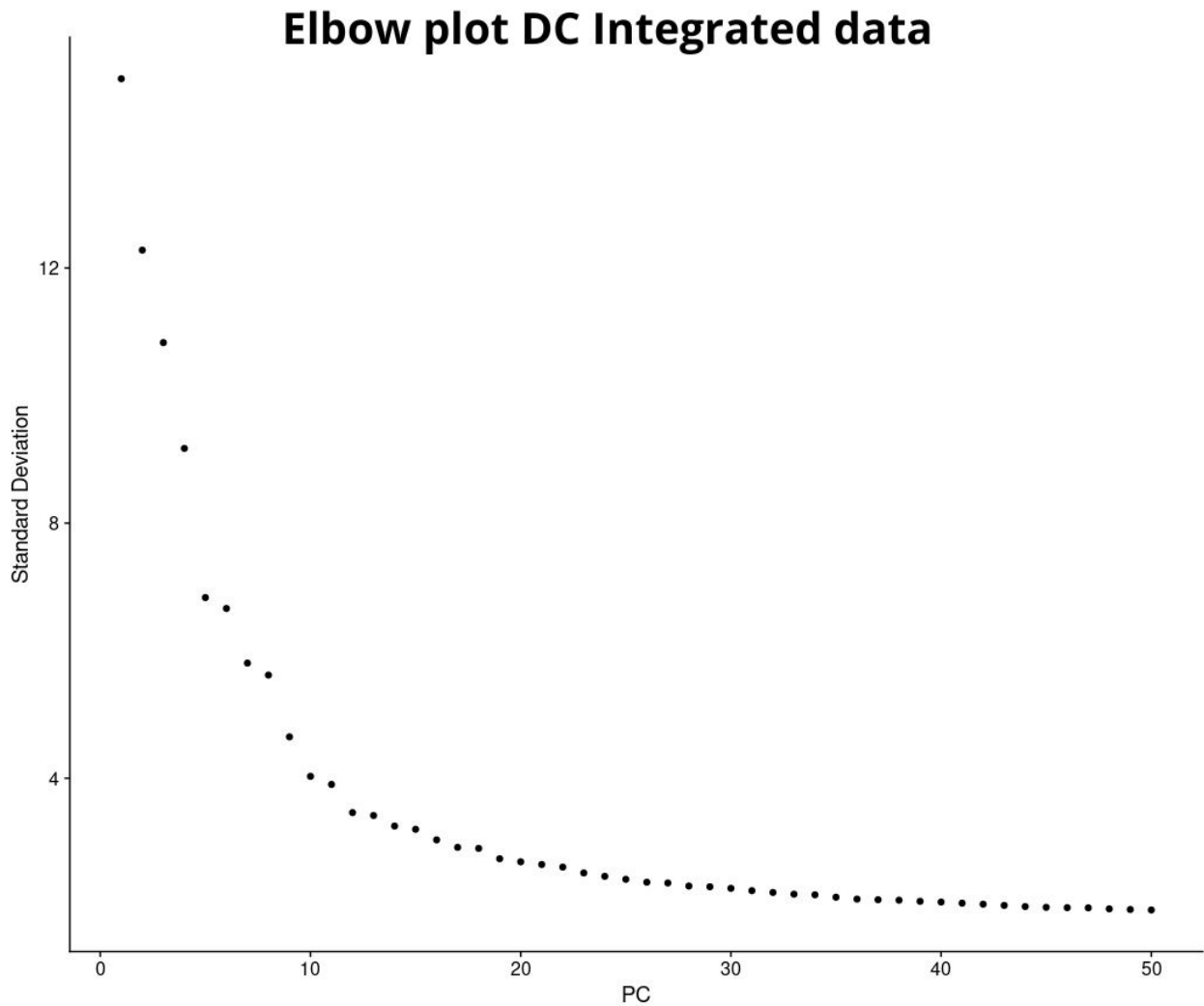


Figure C.9: Elbow plot Dendritic cells

Código C.5: R studio code extract – Dendritic cell non-linear reduction

```

1 #- UMAP Reduction -#
2 samplesIntegFltDC <- RunUMAP(samplesIntegFltDC , dims=1:40, verbose=↵
  ↵ F)
3
4 #UMAP by Condition
5 DimPlot(samplesIntegFltDC , reduction = "umap" , group.by = '↵
  ↵ Treatment' , pt.size = 1.5) + ggtitle("Integrated DC UMAP ↵
  ↵ divided by Condition") + scale_color_manual(labels = c("↵
  ↵ Untreated" , "Treated") , values = c("#F8766D" , "#00bfc4"))
6
7 #UMAP by Sample
8 DimPlot(samplesIntegFltDC , reduction = "umap" , group.by = 'orig.↵
  ↵ ident' , pt.size = 1.5) + ggtitle("Integrated DC UMAP divided↵

```

```
↪ by Sample") + scale_color_manual(labels = c("Untreated1", "↪  
↪ Untreated2", "Treated1", "Treated2"), values = my_colors)
```

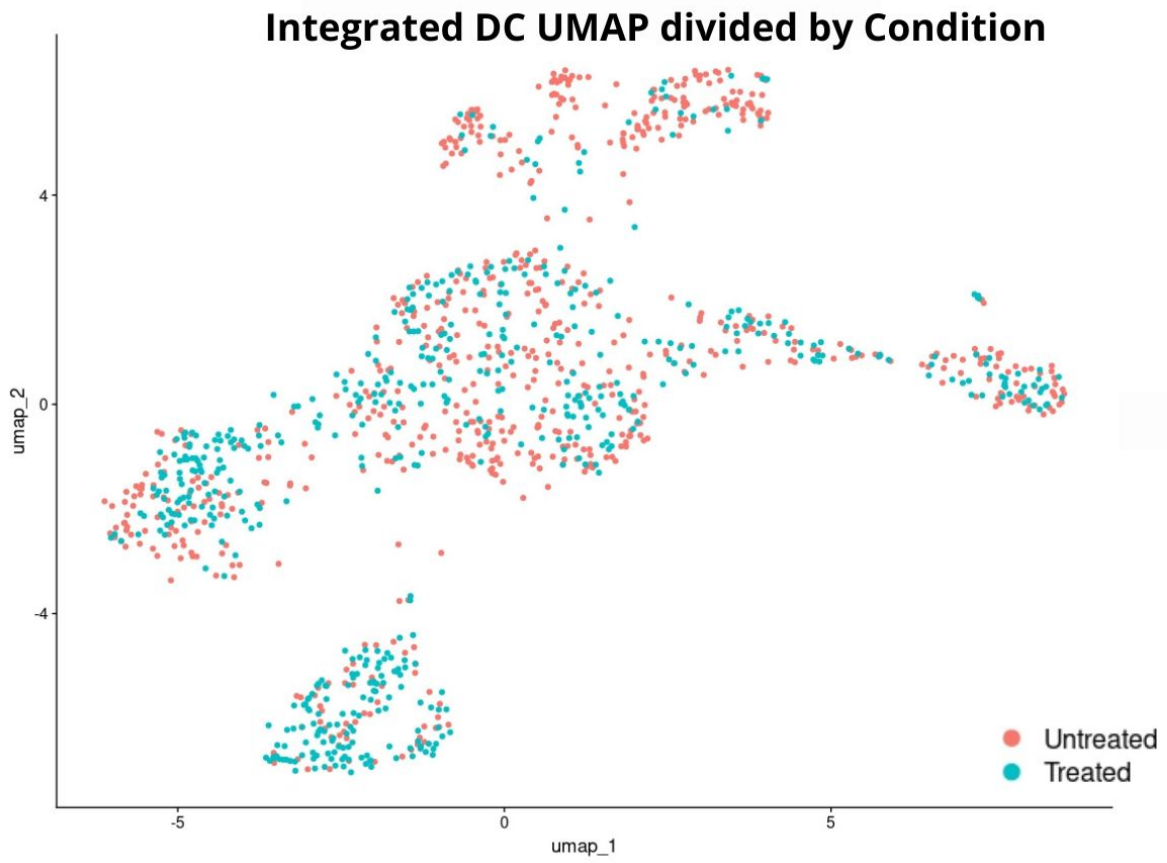


Figure C.10: UMAP of Dendritic cells divided by Condition

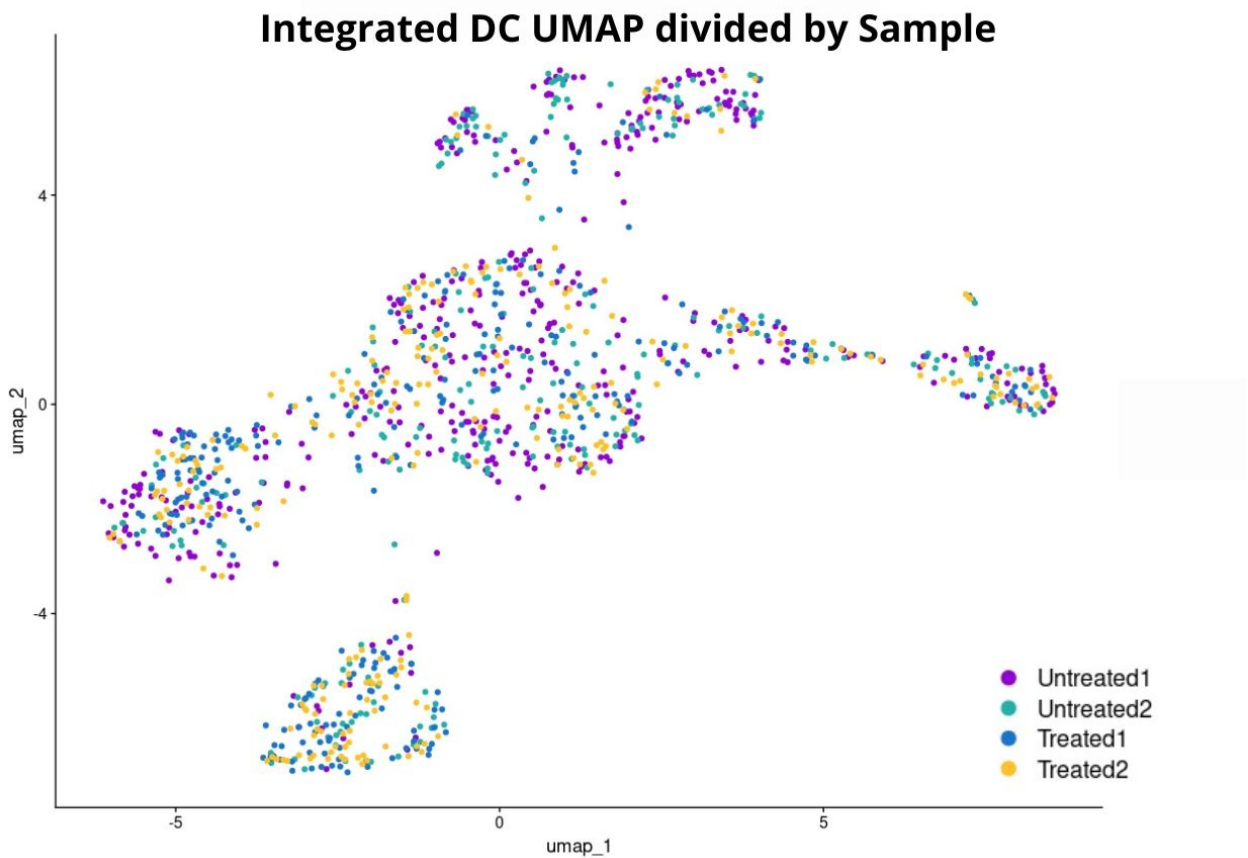


Figure C.11: UMAP of Dendritic cells divided by Sample

Código C.6: R studio code extract – Manual filter of low feature cells

```

1 #--- Get rid of low feature cells manually ---#
2 samplesIntegFltDC <- subset(samplesIntegFltDC , idents = "Low ↵
  ↵ features" , invert = TRUE)
3 FeaturePlot(samplesIntegFltDC , features = 'nFeature_SCT' , min.↵
  ↵ cutoff = 0 , pt.size = 1.5 , cols = c("royalblue" , "brown2"))
4
5 plot <- FeaturePlot(samplesIntegFltDC , features = 'nFeature_SCT' , ↵
  ↵ min.cutoff = 0 , pt.size = 1)
6 select.cells <- CellSelector(plot = plot)
7 Idents(samplesIntegFltDC , cells = select.cells) <- "Low features"

```

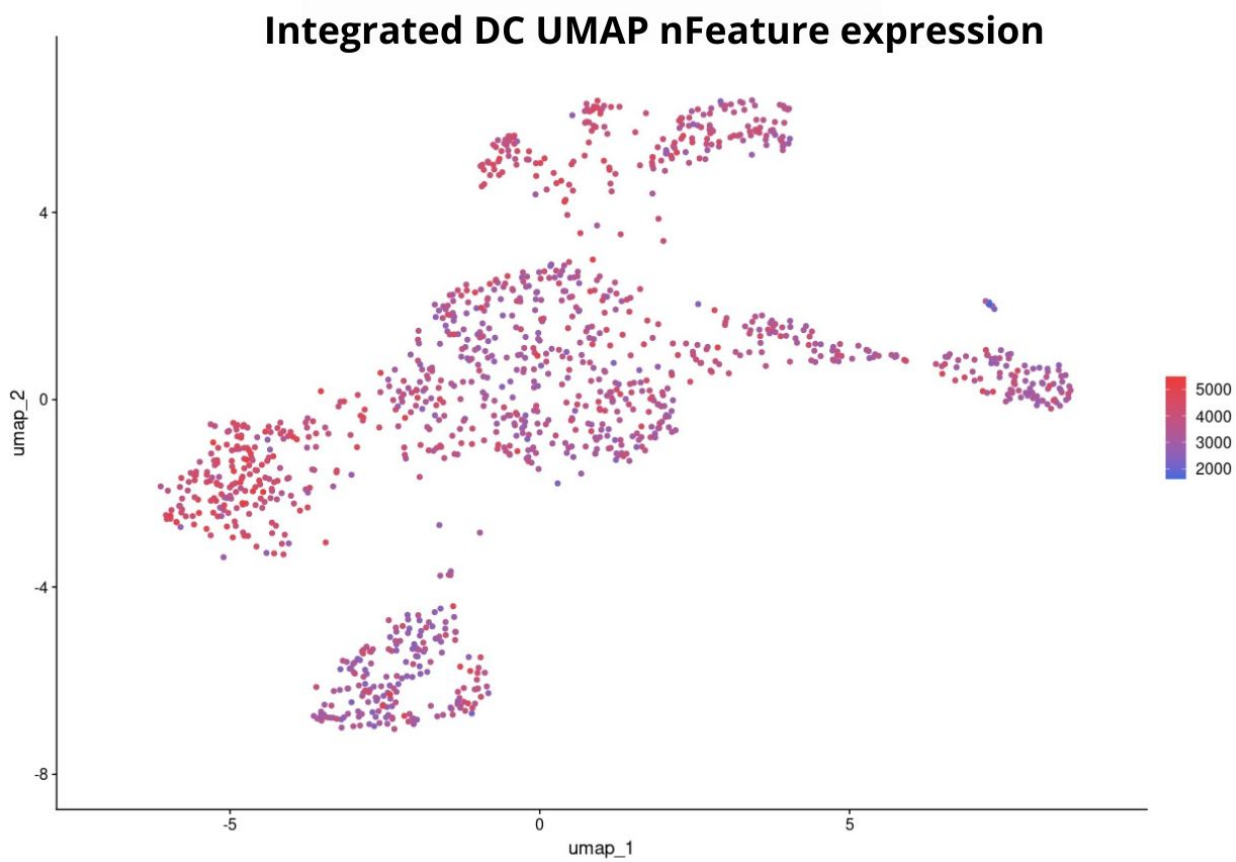


Figure C.12: UMAP of Dendritic cells level of feature expression

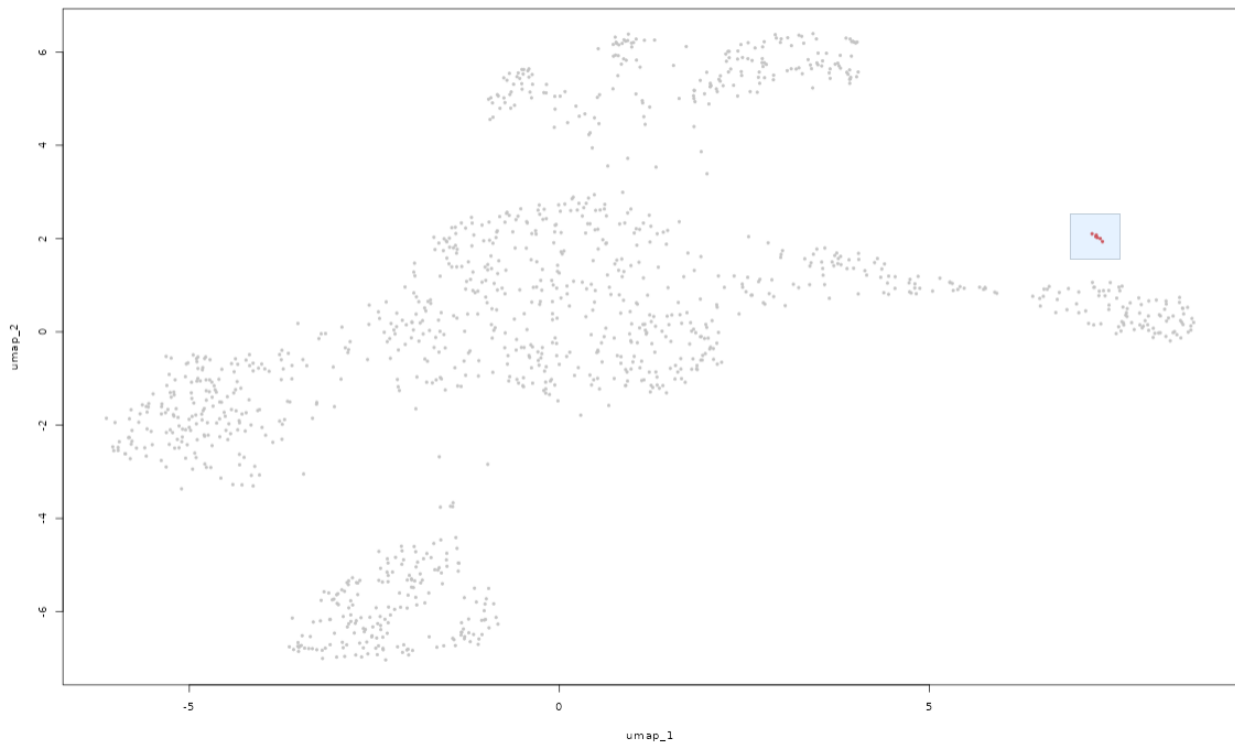


Figure C.13: Manual selection of low features cells

Código C.7: R studio code extract – Dendritic cell Integration

```

1 #--- Re-Integration ---#
2
3 samplesDC.list <- list(Untreated1, Untreated2, Treated1, Treated2)
  ↪ )
4
5 #- Normalice samples -#
6 samplesDC.list <- lapply(X= samplesDC.list , FUN = function(x){
7   x <- SCTransform(object = x, vars.to.regress = "percent.mt",
  ↪   ↪ verbose = FALSE, variable.features.n = 5000)})
8
9 #- Select integration features and prepare for integration -#
10 samplesDC.features <- SelectIntegrationFeatures(object.list =
  ↪ samplesDC.list )
11 samplesDC.list <- PrepSCTIntegration(object.list = samplesDC.list ,
  ↪   ↪ anchor.features = samplesDC.features)
12
13 #- Select integration anchors -#
14 samplesDC.anchors <- FindIntegrationAnchors(object.list =
  ↪ samplesDC.list , normalization.method = "SCT", anchor.
  ↪   ↪ features = samplesDC.features)

```

```

15 save(samplesDC.anchors , file=file.path("~/", "samplesDC.anchors.↵
    ↵ rdata"))
16
17 #- Integrate -#
18 samplesIntegFltDC <- IntegrateData(anchorset = samplesDC.anchors , ↵
    ↵ normalization.method = "SCT")
19 saveRDS(samplesIntegFltDC , "samplesIntegFltDC.rds")

```

Código C.8: R studio code extract – Dendritic cell linear reduction

```

1 #- PCA reduction -#
2 samplesIntegFltDC <- RunPCA(samplesIntegFltDC , npcs = 50, verbose=F↵
    ↵ )
3
4 #Elbow plot
5 ElbowPlot(samplesIntegFltDC , ndims = 50, reduction = "pca") + ↵
    ↵ ggtitle("Elbow plot Integrated DC data")

```

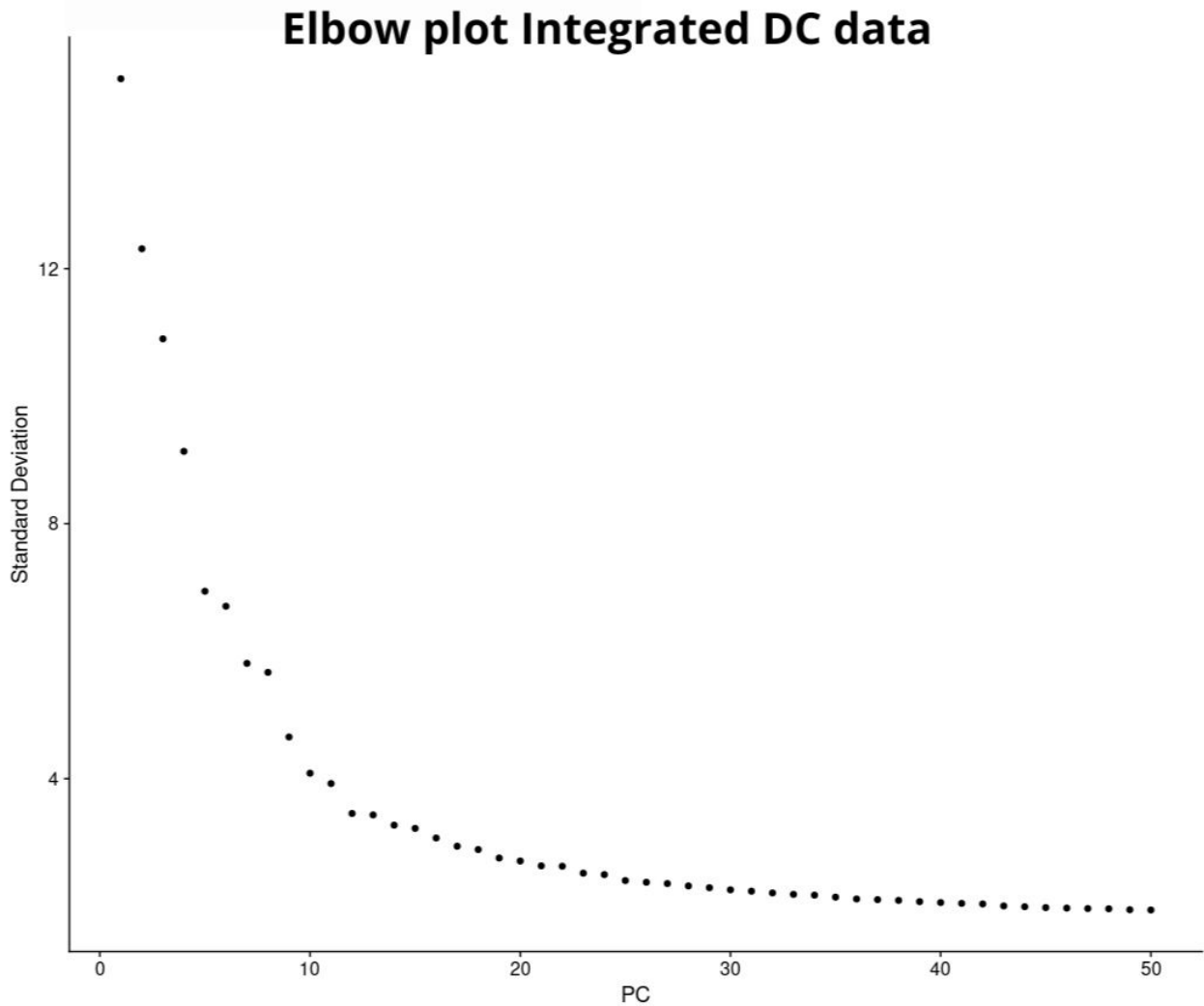


Figure C.14: Elbow plot Dendritic cells

Código C.9: R studio code extract – Dendritic cell non-linear reduction

```

1 #- UMAP Reduction -#
2 samplesIntegFltDC <- RunUMAP(samplesIntegFltDC , dims=1:40, verbose=↵
  ↵ F)
3
4 #UMAP by Condition
5 DimPlot(samplesIntegFltDC , reduction = "umap" , group.by = '↵
  ↵ Treatment' , pt.size = 1.5) + ggtitle("Integrated DC UMAP ↵
  ↵ divided by Condition") + scale_color_manual(labels = c("↵
  ↵ Untreated" , "Treated") , values = c("#F8766D" , "#00bfc4"))
6
7 #UMAP by Sample
8 DimPlot(samplesIntegFltDC , reduction = "umap" , group.by = 'orig.↵
  ↵ ident' , pt.size = 1.5) + ggtitle("Integrated DC UMAP divided↵

```

```
↪ by Sample") + scale_color_manual(labels = c("Untreated1", "↪  
↪ Untreated2", "Treated1", "Treated2"), values = my_colors)
```

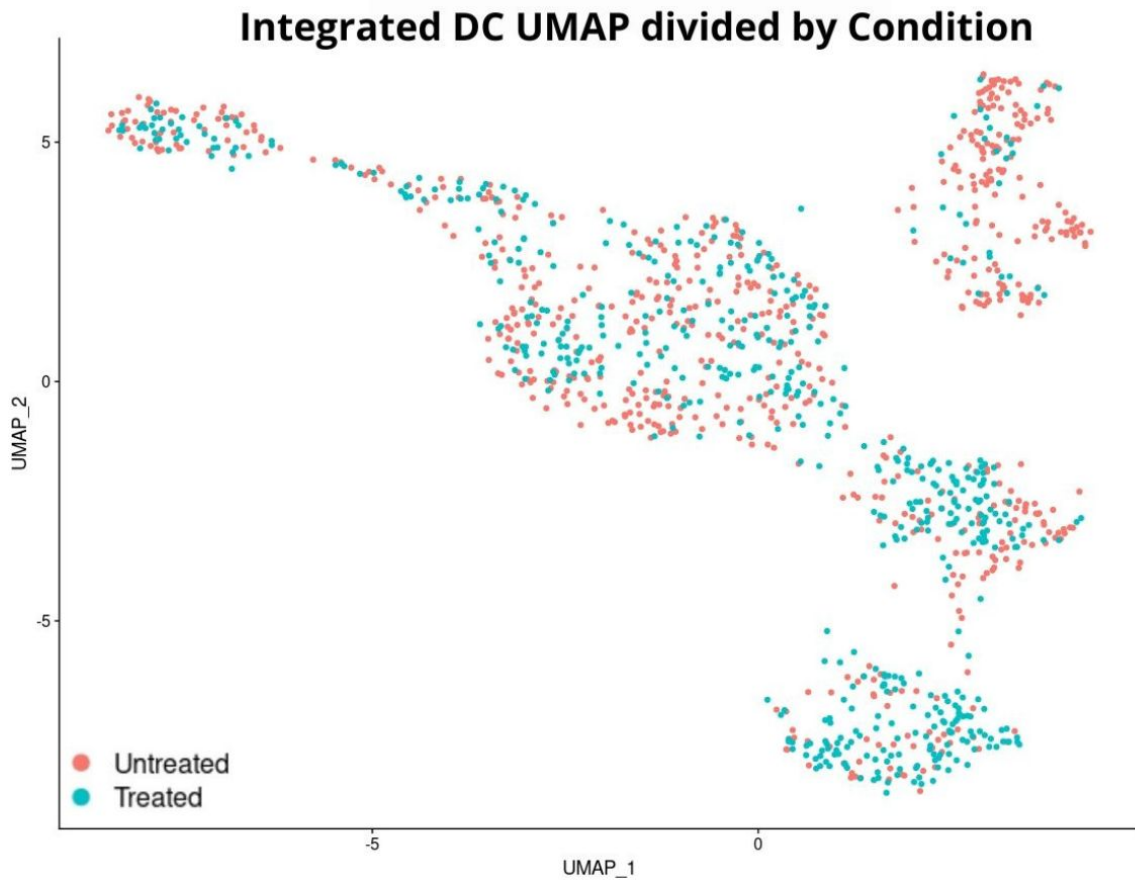


Figure C.15: UMAP of Dendritic cells divided by Condition

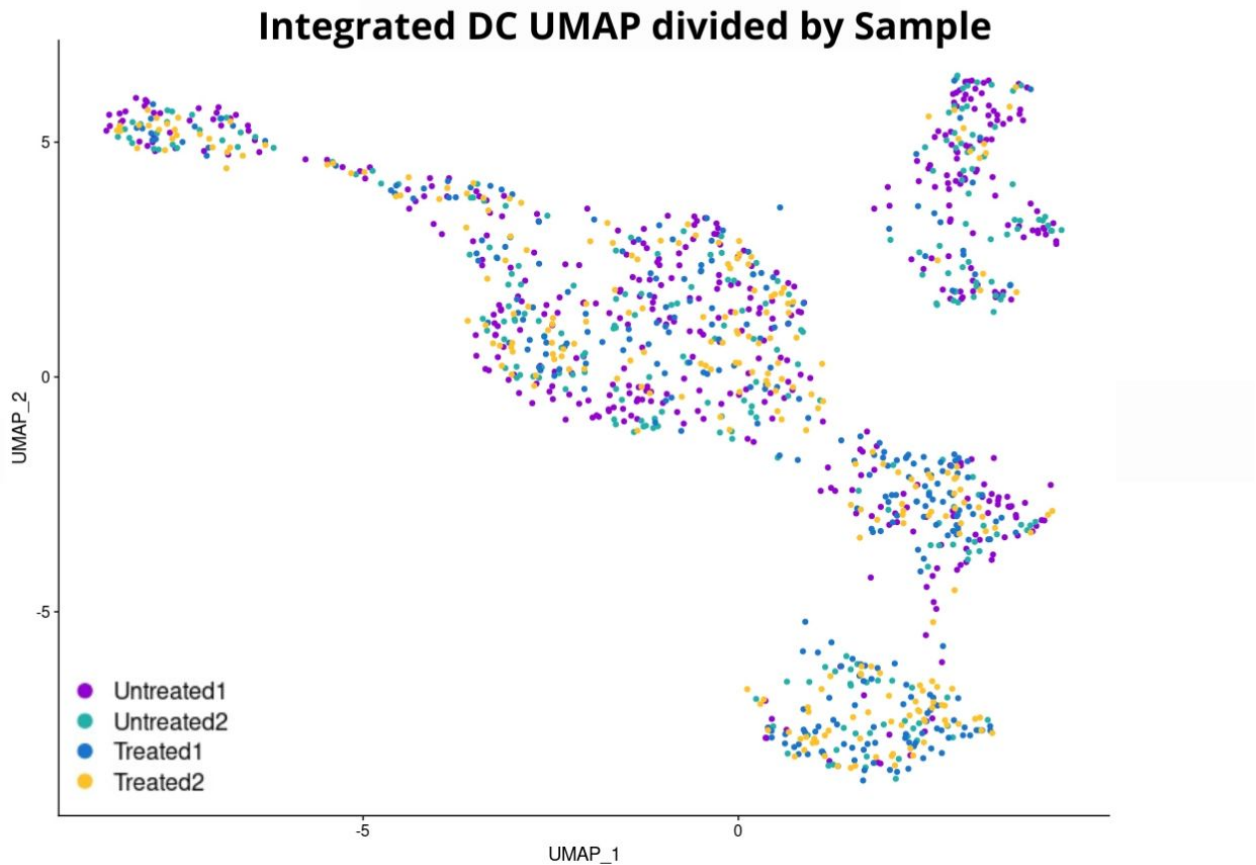


Figure C.16: UMAP of Dendritic cells divided by Sample

Código C.10: R studio code extract – Dendritic cell clustering

```

1 #--- New clustering ---#
2 DefaultAssay(object = samplesIntegFltDC) <- "integrated"
3
4 samplesIntegFltDC <- FindNeighbors(samplesIntegFltDC, dims = 1:50, ↵
  ↵ verbose = FALSE)
5
6 samplesIntegFltDC <- FindClusters(samplesIntegFltDC, resolution = ↵
  ↵ 0)
7 DimPlot(samplesIntegFltDC, reduction = "umap", label = TRUE, pt.↵
  ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0 ↵
  ↵ Dendritic cells')
8
9 samplesIntegFltDC <- FindClusters(samplesIntegFltDC, resolution = ↵
  ↵ 0.1)
10 DimPlot(samplesIntegFltDC, reduction = "umap", label = TRUE, pt.↵
  ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.1 ↵
  ↵ Dendritic cells')
11

```

```

12 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.2)
13 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵
    ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.2 ↵
    ↵ Dendritic cells ')
14
15 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.3)
16 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵
    ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.3 ↵
    ↵ Dendritic cells ')
17
18 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.4)
19 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵
    ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.4 ↵
    ↵ Dendritic cells ')
20
21 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.5)
22 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵
    ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.5 ↵
    ↵ Dendritic cells ')
23
24 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.6)
25 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵
    ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.6 ↵
    ↵ Dendritic cells ')
26
27 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.7)
28 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵
    ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.7 ↵
    ↵ Dendritic cells ')
29
30 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.8)
31 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵
    ↵ size = 1, cols = my_colors) + ggtitle('Resolution 0.8 ↵
    ↵ Dendritic cells ')
32
33 samplesIntegFltDC <- FindClusters(samplesIntegFltDC , resolution = ↵
    ↵ 0.9)
34 DimPlot(samplesIntegFltDC , reduction = "umap" , label = TRUE, pt.↵

```

```
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.9 ↪
    ↪ Dendritic cells')
35
36 samplesIntegFltDC <- FindClusters(samplesIntegFltDC, resolution = ↪
    ↪ 1.0)
37 DimPlot(samplesIntegFltDC, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 1.0 ↪
    ↪ Dendritic cells')
38
39 clustree(samplesIntegFltDC)
```

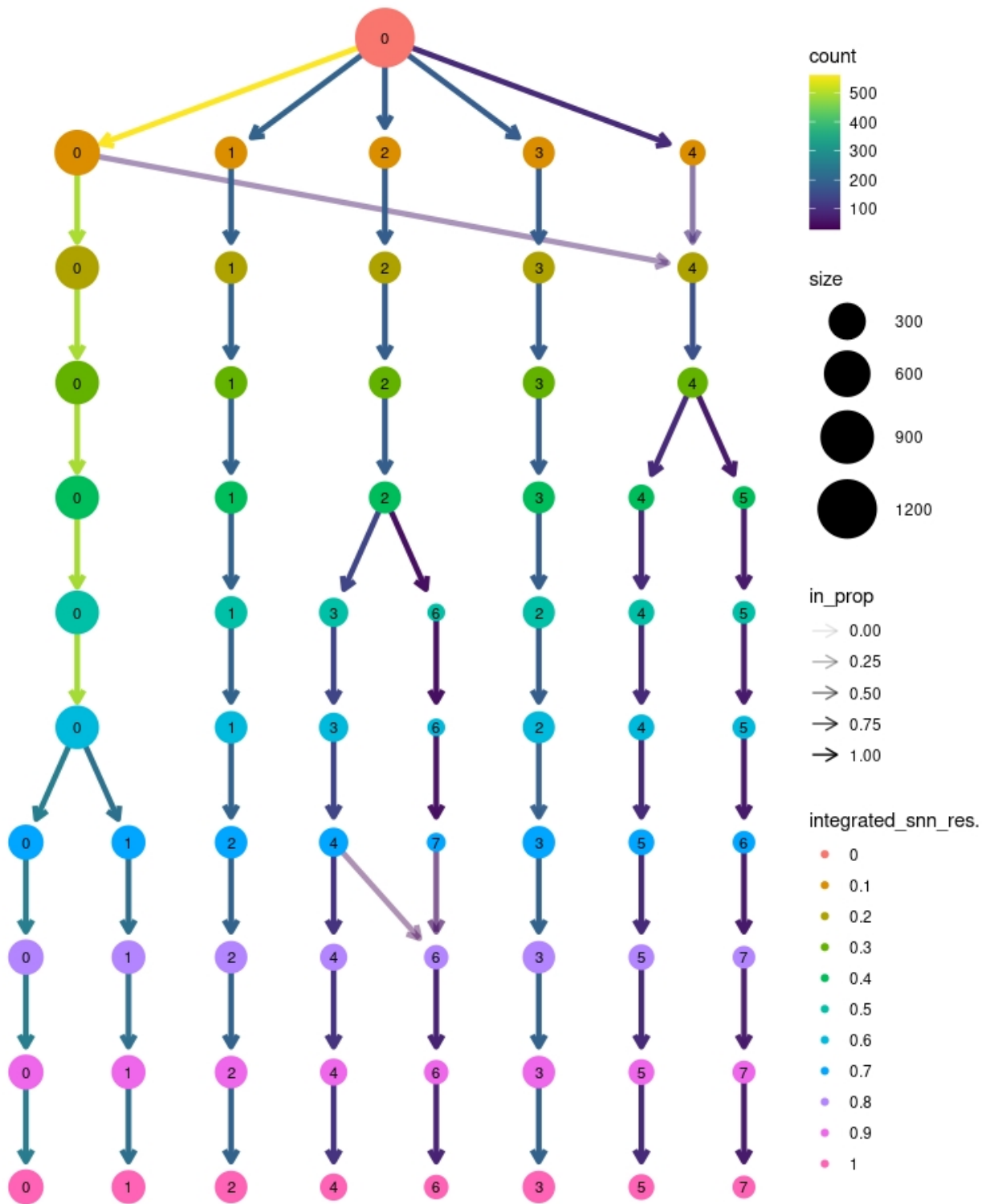


Figure C.17: Dendritic cells Clustree

Código C.11: R studio code extract – Dendritic cell density plot


```

1 #--- Density plot ---#
2 DefaultAssay(object = samplesIntegFltDC) <- "SCT"
3
4 sampList <- SplitObject(samplesIntegFltDC, split.by = "Treatment" <-
  ↪ )
5
6 DC_Untreated <- sampList[[1]]
7 DC_Treated <- sampList[[2]]
8
9 saveRDS(DC_Untreated, "DC_Untreated.rds")
10 saveRDS(DC_Treated, "DC_Treated.rds")
11
12 #- Normal -#
13 p <- DimPlot(samplesIntegFltDC, group.by = "Treatment", reduction = <-
  ↪ "umap")
14
15 #Prepare data
16 data <- data.frame(samplesIntegFltDC@meta.data)
17 data$UMAP1 <- Embeddings(samplesIntegFltDC, "umap")[,1]
18 data$UMAP2 <- Embeddings(samplesIntegFltDC, "umap")[,2]
19
20 #Create a data frame that doesn't contain a "sample" column.
21 data_bg <- data[, -(which(colnames(data)=="Sample"))]
22
23 #Plot
24 density_plot <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
25   geom_point(colour="#00000000") +
26   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins <-
  ↪ =25) +
27   scale_fill_gradientn(colors = c("#4169E100", "royalblue", " <-
  ↪ darkolivegreen3", "goldenrod1", "red")) +
28   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
29   #facet_wrap(~Treatment, ncol=3)+
30   theme_classic() + ggtitle("Density plot") +
31   theme(plot.title = element_text(hjust = 0.5), text = element_ <-
  ↪ text(size = 14))
32
33 density_plot
34
35 #- density 2: Untreated -#
36
37 #Prepare data
38 data <- data.frame(DC_Untreated@meta.data)
39 data$UMAP1 <- Embeddings(DC_Untreated, "umap")[,1]
40 data$UMAP2 <- Embeddings(DC_Untreated, "umap")[,2]

```

```

41
42 #Create a data frame that doesn't contain a "sample" column.
43 data_bg <- data[, -(which(colnames(data)=="Sample"))]
44
45 #Plot
46 density_plot_2 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
47   geom_point(colour="#00000000") +
48   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↵
↵   ↵ =25) +
49   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↵
↵   ↵ darkolivegreen3", "goldenrod1", "red")) +
50   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
51   facet_wrap(~Treatment, ncol=3)+
52   theme_classic() + ggtitle("Untreated Density plot") +
53   theme(plot.title = element_text(hjust = 0.5), text = element_↵
↵   ↵ text(size = 14))
54
55 density_plot_2
56
57 #- density 3: Untreated -#
58
59 #Prepare data
60 data <- data.frame(DC_Treated@meta.data)
61 data$UMAP1 <- Embeddings(DC_Treated, "umap")[,1]
62 data$UMAP2 <- Embeddings(DC_Treated, "umap")[,2]
63
64 #Create a data frame that doesn't contain a "sample" column.
65 data_bg <- data[, -(which(colnames(data)=="Sample"))]
66
67 #Plot
68 density_plot_3 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
69   geom_point(colour="#00000000") +
70   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↵
↵   ↵ =25) +
71   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↵
↵   ↵ darkolivegreen3", "goldenrod1", "red")) +
72   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
73   facet_wrap(~Treatment, ncol=3)+
74   theme_classic() + ggtitle("Treated Density plot") +
75   theme(plot.title = element_text(hjust = 0.5), text = element_↵
↵   ↵ text(size = 14))
76
77 density_plot_3
78
79

```

80 `plot_grid(p, density_plot, density_plot_2, density_plot_3, ncol=2)`

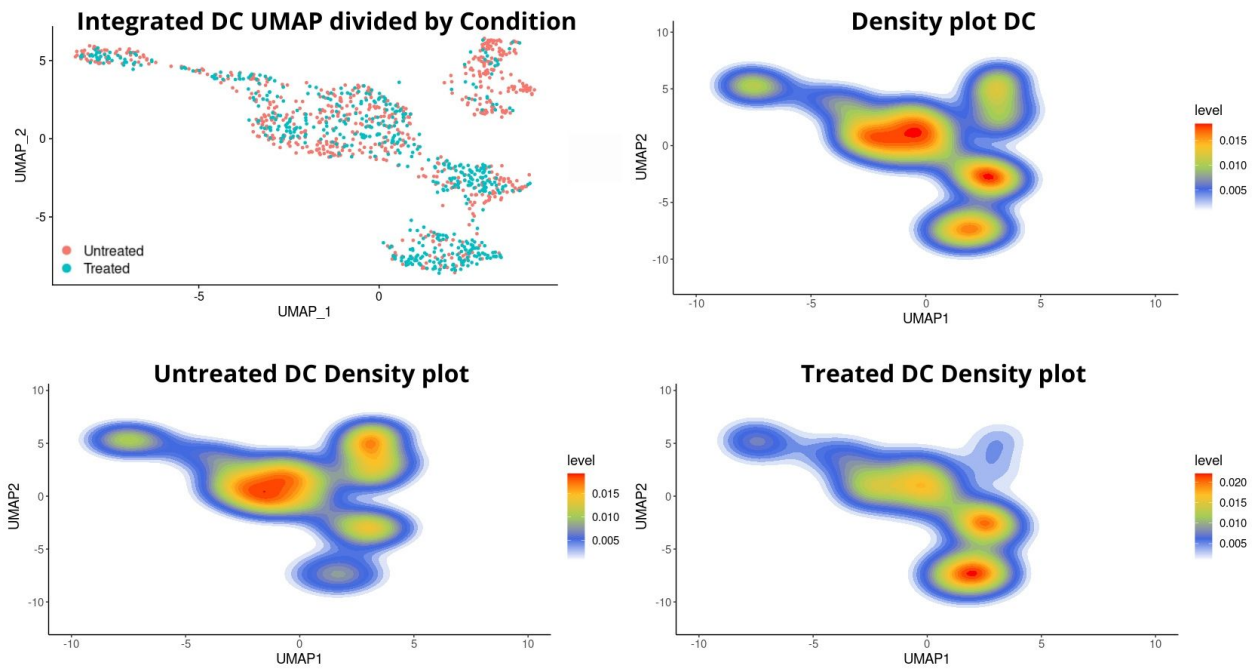


Figure C.18: Dendritic cells density plots

Código C.12: R studio code extract – Dendritic cell signature analysis

```
1 #--- Signature analysis ---#
2 DefaultAssay(object = samplesIntegFltDC) <- "integrated"
3
4 #Chosen resolution:
5 samplesIntegFltDC <- FindClusters(samplesIntegFltDC, resolution = ↵
  ↵ 0.2)
6 DimPlot(samplesIntegFltDC, reduction = "umap", label = FALSE, pt.↵
  ↵ size = 2, cols = my_colors) + ggtitle('Resolution 0.2 ↵
  ↵ Dendritic cells')
```

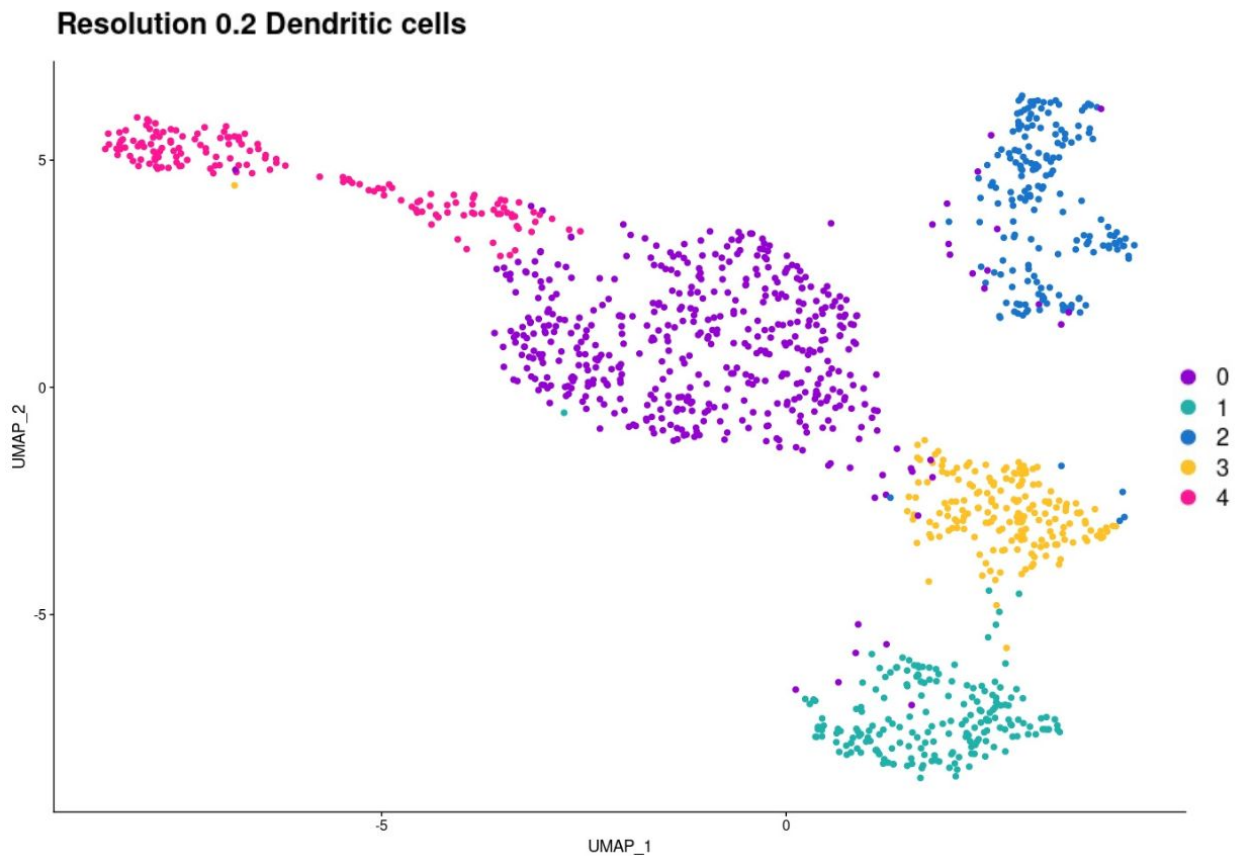


Figure C.19: Integrated Dendritic cells UMAP resolution 0.2

Código C.13: R studio code extract – Dendritic cell signature analysis:
cDC1

```

1 #-- cDC1 --#
2 DefaultAssay(object = samplesIntegFltDC) <- "SCT"
3
4 #Load data
5 data_cDC1<-read.xlsx("DCSignatures.xlsx", sheet = "cDC1")
6
7 Zilionis.mDC1 <- data_cDC1[,1]
8 LiuSpleen.DC1 <- data_cDC1[,2]
9 Maier.DC1 <- data_cDC1[,3]
10 DelPrete.DC1 <- data_cDC1[,4]
11 Marmonti.DC1 <- data_cDC1[,5]
12 Gerhard.cDC1 <- data_cDC1[,6]
13 Villani.DC1CD141CLEC9A <- data_cDC1[,7]
14 Ma.Cluster2 <- data_cDC1[,8]
15 Marciscano.cDC1 <- data_cDC1[,9]
16 Bosteels.NonMigratorycDC1 <- data_cDC1[,11]
17

```

```

18 #Top most repeated genes
19 data_DC1_Top<- c(Zilionis.mDC1, LiuSpleen.DC1, Maier.DC1, DelPrete<-
    ↪ .DC1, Marmonti.DC1, Gerhard.cDC1, Villani.DC1CD141CLEC9A,
20             Ma.Cluster2, Marciscano.cDC1, Bosteels.<-
    ↪ NonMigratorycDC1)
21 sort(table(data_DC1_Top),decreasing=TRUE)[1:20] #at least repeated<-
    ↪ 5 times
22
23 #Gene list
24 Zilionis.mDC1 <- list(Zilionis.mDC1)
25 LiuSpleen.DC1 <- list(LiuSpleen.DC1)
26 Maier.DC1 <- list(Maier.DC1)
27 DelPrete.DC1 <- list(DelPrete.DC1)
28 Marmonti.DC1 <- list(Marmonti.DC1)
29 Gerhard.cDC1 <- list(Gerhard.cDC1)
30 Villani.DC1CD141CLEC9A <- list(Villani.DC1CD141CLEC9A)
31 Ma.Cluster2 <- list(Ma.Cluster2)
32 Marciscano.cDC1 <- list(Marciscano.cDC1)
33 Bosteels.NonMigratorycDC1 <- list(Bosteels.NonMigratorycDC1)
34
35 #Calculate expression level
36 DefaultAssay(object = samplesIntegFltDC) <- "SCT"
37 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Zilionis.mDC1, name = 'Zilionis.mDC1')
38 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = LiuSpleen.DC1, name = 'LiuSpleen.DC1')
39 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Maier.DC1, name = 'Maier.DC1')
40 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = DelPrete.DC1, name = 'DelPrete.DC1')
41 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Marmonti.DC1, name = 'Marmonti.DC1')
42 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Gerhard.cDC1, name = 'Gerhard.cDC1')
43 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Villani.DC1CD141CLEC9A, name = 'Villani.<-
    ↪ DC1CD141CLEC9A')
44 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Ma.Cluster2, name = 'Ma.Cluster2')
45 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Marciscano.cDC1, name = 'Marciscano.cDC1')
46 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
    ↪ features = Bosteels.NonMigratorycDC1, name = 'Bosteels.<-
    ↪ NonMigratorycDC1')
47

```

```

48
49 #Save in new assay
50 samplesIntegFltDC[['cdmodule']] <- CreateAssayObject(data = t(x = ↵
    ↵ FetchData(object = samplesIntegFltDC, vars = c("Zilionis.↵
    ↵ mDC11", "LiuSpleen.DC11", "Maier.DC11", "DelPrete.DC11", "↵
    ↵ Marmonti.DC11", "Gerhard.cDC11", "Villani.DC1CD141CLEC9A1", ↵
    ↵ "Ma.Cluster21", "Marciscano.cDC11", "Bosteels.↵
    ↵ NonMigratorycDC11"))))
51 combined_averages <- AverageExpression(samplesIntegFltDC, return.↵
    ↵ seurat = FALSE, features = c("Zilionis.mDC11", "LiuSpleen.↵
    ↵ DC11", "Maier.DC11", "DelPrete.DC11", "Marmonti.DC11", "↵
    ↵ Gerhard.cDC11", "Villani.DC1CD141CLEC9A1", "Ma.Cluster21", "↵
    ↵ Marciscano.cDC11", "Bosteels.NonMigratorycDC11"), group.by =↵
    ↵ 'integrated_snn_res.0.2', assays = 'cdmodule')
52
53 write.csv(combined_averages, "~/cDC1Signature.csv", row.names=TRUE)
54
55 #Generate normalized data
56 Datos<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↵
    ↵ cDC1Signature")
57
58 data_new <- Datos[, unlist(lapply(Datos, is.numeric))]
59 data_new
60
61 str(data_new)
62
63 data <- t(scale(t(data_new),center=T,scale=T))
64 data1 <- melt(data)
65
66 head(data1)
67
68 colnames(data1) <- c("Signature", "Cluster", "value")
69
70 write.csv(data1, "~/cDC1SignatureNorm.csv", row.names=TRUE)
71
72 data1<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↵
    ↵ cDC1SignatureNorm")
73
74 ggplot(data1, aes(x = Cluster, y = Signature, fill = value)) +
75   geom_tile() +
76   labs(title = "cDC1 Signatures comparison",
77         x = "Cluster",
78         y = "Signature")+
79   guides(fill = guide_colourbar(title = "Expression level"))+
80   scale_fill_gradient2(high = "brown2", mid = "lemonchiffon", low ↵

```

```

↔ = "royalblue") +
81 theme(axis.text=element_text(size=16, face="bold"),
82        axis.title=element_text(size=13,face="bold"), title = ↔
↔ element_text(size=18,face="bold"))

```

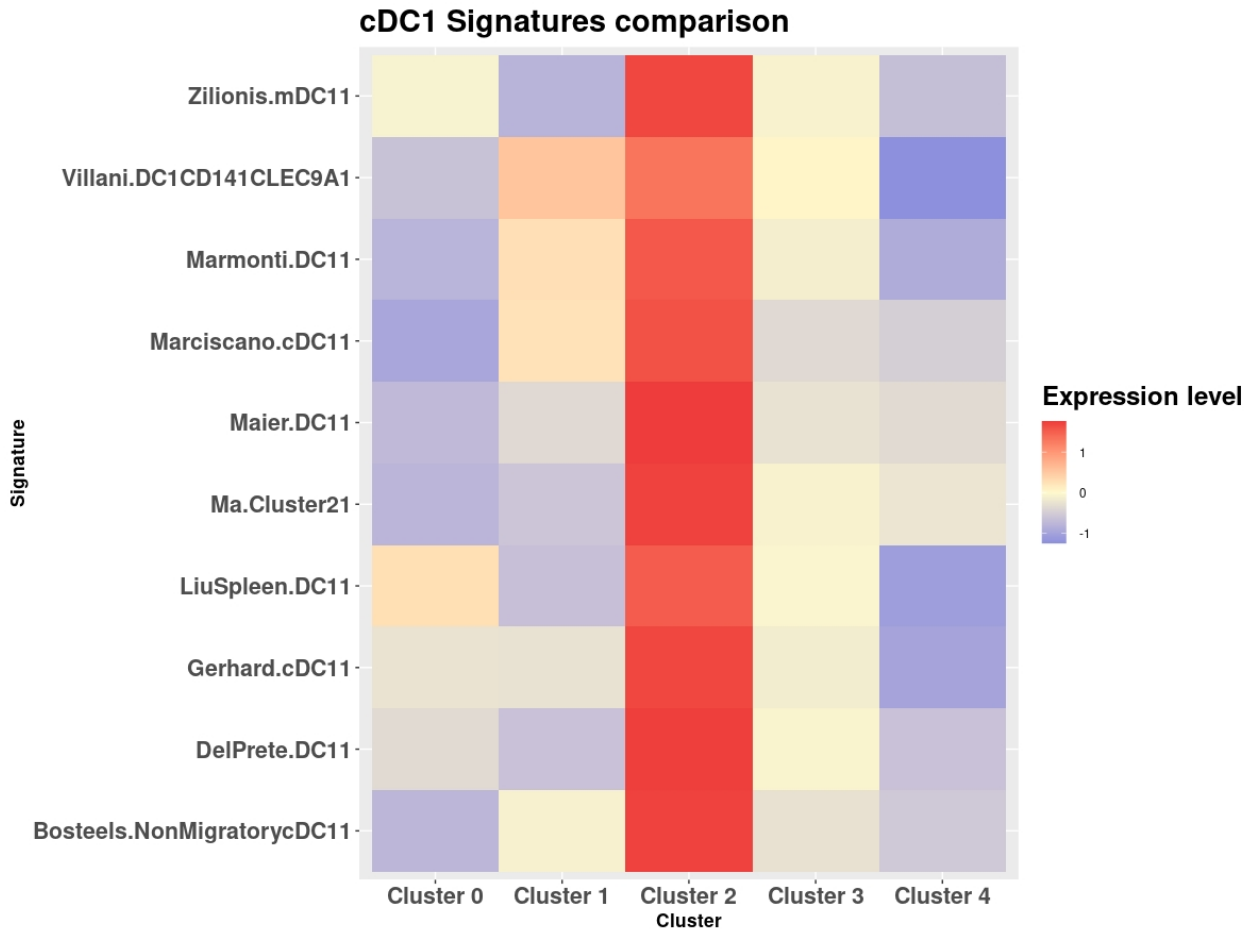


Figure C.20: Heatmap cDC1 Signatures

Código C.14: R studio code extract – Dendritic cell signature analysis: cDC2

```

1 #-- cDC2 --#
2
3 #Load data
4 data_cDC2<-read.xlsx("DCSignatures.xlsx", sheet = "cDC2")
5
6 Zilionis.mDC2 <- data_cDC2[,1]
7 LiuSpleen.DC2 <- data_cDC2[,2]
8 Maier.DC2 <- data_cDC2[,3]
9 DelPrete.DC2 <- data_cDC2[,4]
10 Marmonti.DC2 <- data_cDC2[,5]

```

```

11 Gerhard.cDC2 <- data_cDC2[,6]
12 Gerhard.cDC2MoDC <- data_cDC2[,7]
13 Villani.DC2CD1CA <- data_cDC2[,8]
14 Villani.DC3CD1CB <- data_cDC2[,9]
15 Ma.Cluster3 <- data_cDC2[,10]
16 Marciscano.cDC2 <- data_cDC2[,11]
17 Backer.cDC2A <- data_cDC2[,12]
18 Backer.cDC2B <- data_cDC2[,13]
19 Backer.InfcDC2 <- data_cDC2[,14]
20 Bosteels.NonMigratorycDC2 <- data_cDC2[,15]
21 Bosteels.InfcDC2 <- data_cDC2[,16]
22
23 #Top most repeated genes
24 data_DC2_Top <- c(Zilionis.mDC2, LiuSpleen.DC2, Maier.DC2, ←
  ↪ DelPrete.DC2, Marmonti.DC2, Gerhard.cDC2, Gerhard.cDC2MoDC, ←
  ↪ Villani.DC2CD1CA,
25 Villani.DC3CD1CB, Ma.Cluster3, Marciscano.cDC2, ←
  ↪ Backer.cDC2A, Backer.cDC2B, Backer.InfcDC2, Bosteels.←
  ↪ MigratorycDC2, Bosteels.InfcDC2)
26 sort(table(data_DC2_Top), decreasing=TRUE)[1:20] #at least ←
  ↪ repeated 5 times
27
28 #Gene list
29 Zilionis.mDC2 <- list(Zilionis.mDC2)
30 LiuSpleen.DC2 <- list(LiuSpleen.DC2)
31 Maier.DC2 <- list(Maier.DC2)
32 DelPrete.DC2 <- list(DelPrete.DC2)
33 Marmonti.DC2 <- list(Marmonti.DC2)
34 Gerhard.cDC2 <- list(Gerhard.cDC2)
35 Gerhard.cDC2MoDC <- list(Gerhard.cDC2MoDC)
36 Villani.DC2CD1CA <- list(Villani.DC2CD1CA)
37 Villani.DC3CD1CB <- list(Villani.DC3CD1CB)
38 Ma.Cluster3 <- list(Ma.Cluster3)
39 Marciscano.cDC2 <- list(Marciscano.cDC2)
40 Backer.cDC2A <- list(Backer.cDC2A)
41 Backer.cDC2B <- list(Backer.cDC2B)
42 Backer.InfcDC2 <- list(Backer.InfcDC2)
43 Bosteels.NonMigratorycDC2 <- list(Bosteels.NonMigratorycDC2)
44 Bosteels.InfcDC2 <- list(Bosteels.InfcDC2)
45
46 #Calculate expression level
47 DefaultAssay(object = samplesIntegFltDC) <- "SCT"
48 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
  ↪ features = Zilionis.mDC2, name = 'Zilionis.mDC2')
49 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←

```



```

    ↪ features = LiuSpleen.DC2, name = 'LiuSpleen.DC2')
50 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Maier.DC2, name = 'Maier.DC2')
51 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = DelPrete.DC2, name = 'DelPrete.DC2')
52 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Marmonti.DC2, name = 'Marmonti.DC2')
53 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Gerhard.cDC2, name = 'Gerhard.cDC2')
54 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Gerhard.cDC2MoDC, name = 'Gerhard.cDC2MoDC')
55 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Villani.DC2CD1CA, name = 'Villani.DC2CD1CA')
56 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Villani.DC3CD1CB, name = 'Villani.DC3CD1CB')
57 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Ma.Cluster3, name = 'Ma.Cluster3')
58 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Marciscano.cDC2, name = 'Marciscano.cDC2')
59 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Backer.cDC2A, name = 'Backer.cDC2A')
60 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Backer.cDC2B, name = 'Backer.cDC2B')
61 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Backer.InfcDC2, name = 'Backer.InfcDC2')
62 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Bosteels.NonMigratorycDC2, name = 'Bosteels.↪
    ↪ NonMigratorycDC2')
63 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features =Bosteels.InfcDC2, name = 'Bosteels.InfcDC2')
64
65 #Save in new assay
66 samplesIntegFltDC[['cdmodule']] <- CreateAssayObject(data = t(x = ↪
    ↪ FetchData(object = samplesIntegFltDC, vars = c("Zilionis.↪
    ↪ mDC21", "LiuSpleen.DC21", "Maier.DC21", "DelPrete.DC21", "↪
    ↪ Marmonti.DC21", "Gerhard.cDC21", "Gerhard.cDC2MoDC1", "↪
    ↪ Villani.DC2CD1CA1", "Villani.DC3CD1CB1", "Ma.Cluster31", "↪
    ↪ Marciscano.cDC21", "Backer.cDC2A1", "Backer.cDC2B1", "Backer↪
    ↪ .InfcDC21", "Bosteels.NonMigratorycDC21", "Bosteels.InfcDC21↪
    ↪ "))))
67 combined_averages <- AverageExpression(samplesIntegFltDC, return.↪
    ↪ seurat = FALSE, features = c("Zilionis.mDC21", "LiuSpleen.↪
    ↪ DC21", "Maier.DC21", "DelPrete.DC21", "Marmonti.DC21", "↪
    ↪ Gerhard.cDC21", "Gerhard.cDC2MoDC1", "Villani.DC2CD1CA1", "↪
    ↪ Villani.DC3CD1CB1", "Ma.Cluster31", "Marciscano.cDC21", "↪

```

```

↪ Backer.cDC2A1", "Backer.cDC2B1", "Backer.InfcDC21", "↪
↪ Bosteels.NonMigratorycDC21", "Bosteels.InfcDC21"), group.by ↪
↪ = 'integrated_snn_res.0.2', assays = 'cdmodule')
68
69 write.csv(combined_averages, "~/cDC2Signature.csv", row.names=TRUE)
70
71 #Generate normalized data
72 Datos<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↪
↪ cDC2Signature")
73
74 data_new <- Datos[ , unlist(lapply(Datos, is.numeric))]
75 data_new
76
77 str(data_new)
78
79 data <- t(scale(t(data_new), center=T, scale=T))
80 data1 <- melt(data)
81
82 head(data1)
83
84 colnames(data1) <- c("Signature", "Cluster", "value")
85
86 write.csv(data1, "~/cDC2SignatureNorm.csv", row.names=TRUE)
87
88 data1<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↪
↪ cDC2SignatureNorm")
89
90 ggplot(data1, aes(x = Cluster, y = Signature, fill = value)) +
91   geom_tile() +
92   labs(title = "cDC2 Signatures comparison",
93        x = "Cluster",
94        y = "Signature")+
95   guides(fill = guide_colourbar(title = "Expression level"))+
96   scale_fill_gradient2(high = "brown2", mid = "lemonchiffon", low ↪
↪ = "royalblue") +
97   theme(axis.text=element_text(size=16, face="bold"),
98         axis.title=element_text(size=13,face="bold"), title = ↪
↪ element_text(size=18,face="bold"))

```

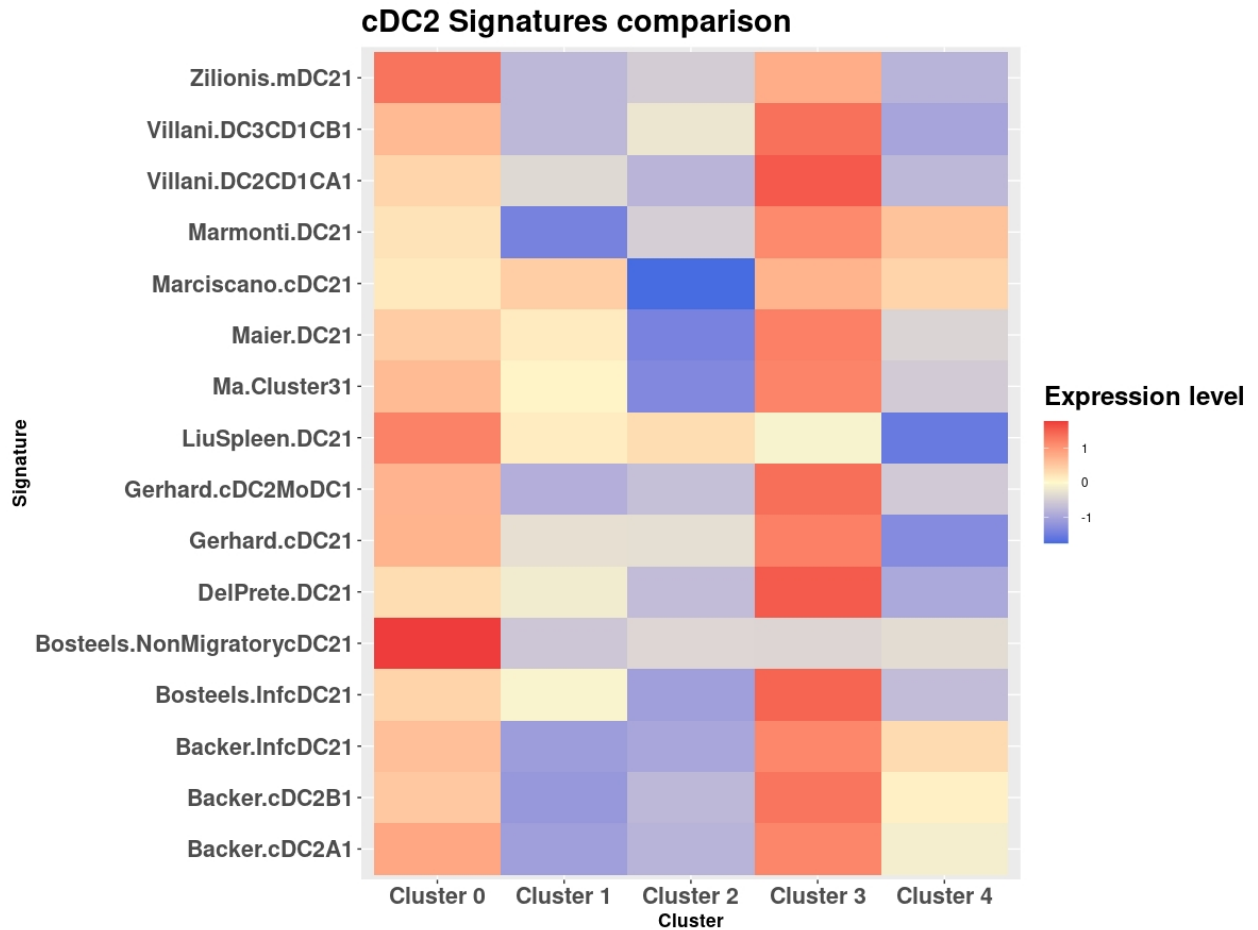


Figure C.21: Heatmap cDC2 Signatures

Código C.15: R studio code extract – Dendritic cell signature analysis: DC3

```

1 #-- DC3 --#
2
3 #Load data
4 data_DC3<-read.xlsx("DCSignatures.xlsx", sheet = "DC3")
5
6 Zilionis.mDC3 <- data_DC3[,1]
7 LiuBlood.CCR7DC <- data_DC3[,2]
8 LiuSpleen.CCR7DC <- data_DC3[,3]
9 Maier.mregDC <- data_DC3[,4]
10 DelPrete.mregDC <- data_DC3[,5]
11 Marmonti.mregDC <- data_DC3[,6]
12 Gerhard.DC3 <- data_DC3[,7]
13 Marciscano.DC3 <- data_DC3[,8]
14
15 #Top most repeated genes
16 data_DC3_Top<- c(Zilionis.mDC3, LiuBlood.CCR7DC, LiuSpleen.CCR7DC, ←

```

```

    ↪ Maier.mregDC, DelPrete.mregDC, Marmonti.mregDC, Gerhard.DC3↪
    ↪ , Marciscano.DC3)
17 sort(table(data_DC3_Top), decreasing=TRUE)[1:20] #at least repeated↪
    ↪ 5 times
18
19 #Gene list
20 Zilionis.mDC3 <- list(Zilionis.mDC3)
21 LiuBlood.CCR7DC <- list(LiuBlood.CCR7DC)
22 LiuSpleen.CCR7DC <- list(LiuSpleen.CCR7DC)
23 Maier.mregDC <- list(Maier.mregDC)
24 DelPrete.mregDC <- list(DelPrete.mregDC)
25 Marmonti.mregDC <- list(Marmonti.mregDC)
26 Gerhard.DC3 <- list(Gerhard.DC3)
27 Marciscano.DC3 <- list(Marciscano.DC3)
28
29 #Calculate expression level
30 DefaultAssay(object = samplesIntegFltDC) <- "SCT"
31 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Zilionis.mDC3, name = 'Zilionis.mDC3')
32 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = LiuBlood.CCR7DC, name = 'LiuBlood.CCR7DC')
33 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = LiuSpleen.CCR7DC, name = 'LiuSpleen.CCR7DC')
34 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Maier.mregDC, name = 'Maier.mregDC')
35 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = DelPrete.mregDC, name = 'DelPrete.mregDC')
36 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Marmonti.mregDC, name = 'Marmonti.mregDC')
37 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Gerhard.DC3, name = 'Gerhard.DC3')
38 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ↪
    ↪ features = Marciscano.DC3, name = 'Marciscano.DC3')
39
40 #Save in new assay
41 samplesIntegFltDC[['cdmodule']] <- CreateAssayObject(data = t(x = ↪
    ↪ FetchData(object = samplesIntegFltDC, vars = c("Zilionis.↪
    ↪ mDC31", "LiuBlood.CCR7DC1", "LiuSpleen.CCR7DC1", "Maier.↪
    ↪ mregDC1", "DelPrete.mregDC1", "Marmonti.mregDC1", "Gerhard.↪
    ↪ DC31", "Marciscano.DC31"))))
42 combined_averages <- AverageExpression(samplesIntegFltDC, return.↪
    ↪ seurat = FALSE, features = c("Zilionis.mDC31", "LiuBlood.↪
    ↪ CCR7DC1", "LiuSpleen.CCR7DC1", "Maier.mregDC1", "DelPrete.↪
    ↪ mregDC1", "Marmonti.mregDC1", "Gerhard.DC31", "Marciscano.↪
    ↪ DC31"), group.by = 'integrated_snn_res.0.2', assays = '↪

```

```

    ↪ cdmodule')
43
44 write.csv(combined_averages, "~/DC3Signature.csv", row.names=TRUE)
45
46 #Generate normalized data
47 Datos<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↪
    ↪ DC3Signature")
48
49 data_new <- Datos[ , unlist(lapply(Datos, is.numeric))]
50 data_new
51
52 str(data_new)
53
54 data <- t(scale(t(data_new),center=T,scale=T))
55 data1 <- melt(data)
56
57 head(data1)
58
59 colnames(data1) <- c("Signature", "Cluster", "value")
60
61 write.csv(data1, "~/DC3SignatureNorm.csv", row.names=TRUE)
62
63 data1<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↪
    ↪ DC3SignatureNorm")
64
65 ggplot(data1, aes(x = Cluster, y = Signature, fill = value)) +
66   geom_tile() +
67   labs(title = "DC3 Signatures comparison",
68        x = "Cluster",
69        y = "Signature")+
70   guides(fill = guide_colourbar(title = "Expression level"))+
71   scale_fill_gradient2(high = "brown2", mid = "lemonchiffon", low ↪
    ↪ = "royalblue") +
72   theme(axis.text=element_text(size=16, face="bold"),
73         axis.title=element_text(size=13,face="bold"), title = ↪
    ↪ element_text(size=18,face="bold"))

```

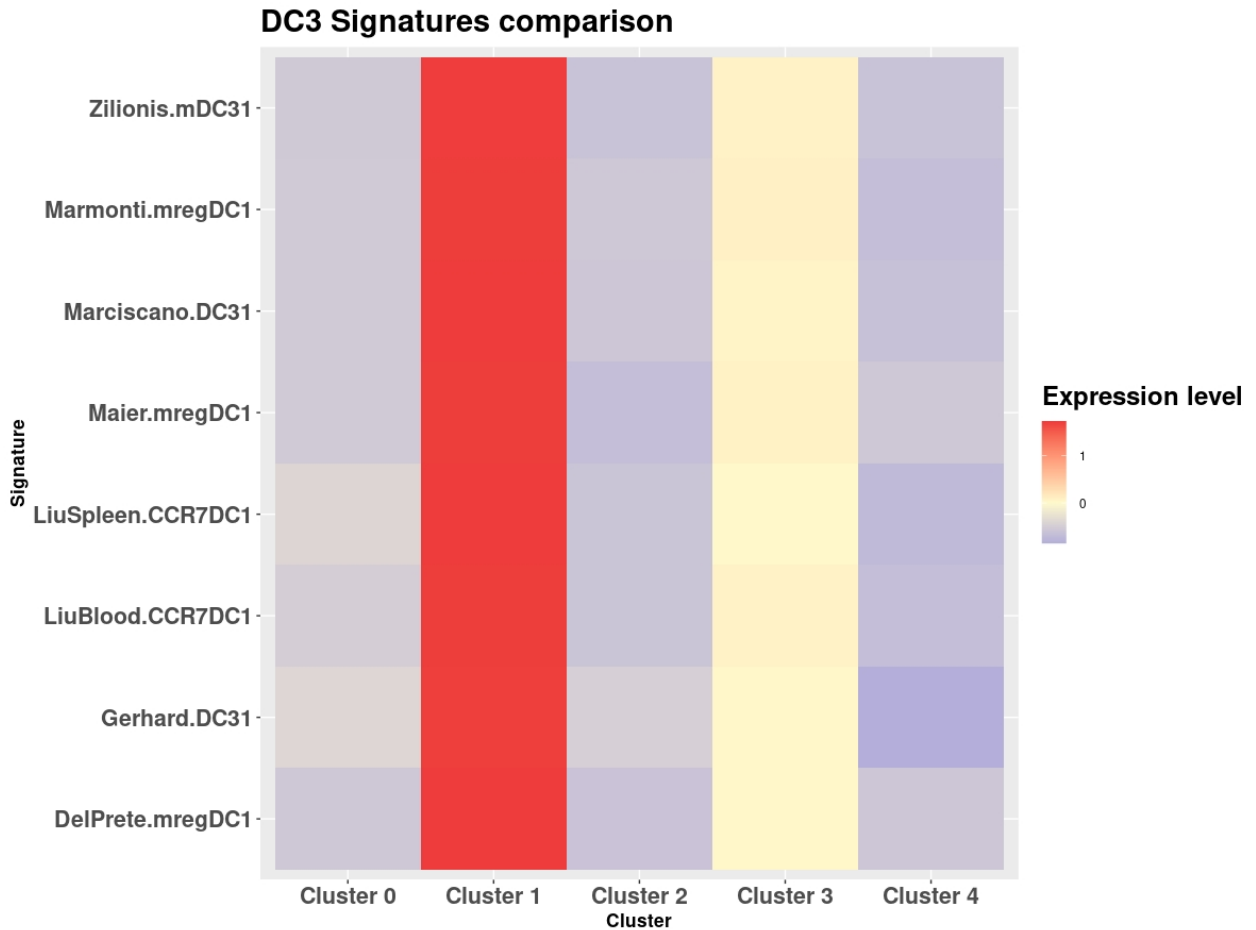


Figure C.22: Heatmap DC3 Signatures

Código C.16: R studio code extract – Dendritic cell signature analysis: pDC

```

1 #-- pDC --#
2
3 #Load data
4 data_pDC<-read.xlsx("DCSignatures.xlsx", sheet = "pDC")
5
6 Zilionis.mpDC <- data_pDC[,1]
7 LiuBM.pDC <- data_pDC[,2]
8 LiuBlood.pDC <- data_pDC[,3]
9 LiuSpleen.pDC <- data_pDC[,4]
10 Maier.pDC <- data_pDC[,5]
11 DelPrete.pDC <- data_pDC[,6]
12 Gerhard.pDC <- data_pDC[,8]
13 Villani.DC6pDC <- data_pDC[,9]
14 Marciscano.pDC <- data_pDC[,10]
15 Bosteels.pDC <- data_pDC[,11]
16

```

```

17 #Top most repeated genes
18 data_pDC_Top<- c(Zilionis.mpDC, LiuBM.pDC, LiuBlood.pDC, LiuSpleen←
    ↪ .pDC, Maier.pDC, DelPrete.pDC, Gerhard.pDC, Villani.DC6pDC, ←
    ↪ Marciscano.pDC, Bosteels.pDC)
19 sort(table(data_pDC_Top),decreasing=TRUE)[1:20] #at least repeated←
    ↪ 6 times
20
21 #Gene list
22 Zilionis.mpDC <- list(Zilionis.mpDC)
23 LiuBM.pDC <- list(LiuBM.pDC)
24 LiuBlood.pDC <- list(LiuBlood.pDC)
25 LiuSpleen.pDC <- list(LiuSpleen.pDC)
26 Maier.pDC <- list(Maier.pDC)
27 DelPrete.pDC <- list(DelPrete.pDC)
28 Gerhard.pDC <- list(Gerhard.pDC)
29 Villani.DC6pDC <- list(Villani.DC6pDC)
30 Marciscano.pDC <- list(Marciscano.pDC)
31 Bosteels.pDC <- list(Bosteels.pDC)
32
33 #Calculate expression level
34 DefaultAssay(object = samplesIntegFltDC) <- "SCT"
35 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = Zilionis.mpDC, name = 'Zilionis.mpDC')
36 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = LiuBM.pDC, name = 'LiuBM.pDC')
37 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = LiuBlood.pDC, name = 'LiuBlood.pDC')
38 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = LiuSpleen.pDC, name = 'LiuSpleen.pDC')
39 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = Maier.pDC, name = 'Maier.pDC')
40 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = DelPrete.pDC, name = 'DelPrete.pDC')
41 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = Gerhard.pDC, name = 'Gerhard.pDC')
42 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = Villani.DC6pDC, name = 'Villani.DC6pDC')
43 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features = Marciscano.pDC, name = 'Marciscano.pDC')
44 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, ←
    ↪ features =Bosteels.pDC, name = 'Bosteels.pDC')
45
46 #Save in new assay
47 samplesIntegFltDC[['cdmodule']] <- CreateAssayObject(data = t(x = ←
    ↪ FetchData(object = samplesIntegFltDC, vars = c("Zilionis.←

```

```

    ↪ mpDC1", "LiuBM.pDC1", "LiuBlood.pDC1", "LiuSpleen.pDC1", "↪
    ↪ Maier.pDC1", "DelPrete.pDC1", "Gerhard.pDC1", "Villani.↪
    ↪ DC6pDC1", "Marciscano.pDC1", "Bosteels.pDC1"))))
48 combined_averages <- AverageExpression(samplesIntegFltDC, return.↪
    ↪ seurat = FALSE, features = c("Zilionis.mpDC1", "LiuBM.pDC1", ↪
    ↪ "LiuBlood.pDC1", "LiuSpleen.pDC1", "Maier.pDC1", "DelPrete.↪
    ↪ pDC1", "Gerhard.pDC1", "Villani.DC6pDC1", "Marciscano.pDC1", ↪
    ↪ "Bosteels.pDC1"), group.by = 'integrated_snn_res.0.2', ↪
    ↪ assays = 'cdmodule')
49
50 write.csv(combined_averages, "~/pDCSignature.csv", row.names=TRUE)
51
52 #Generate normalized data
53 Datos<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↪
    ↪ pDCSignature")
54
55 data_new <- Datos[ , unlist(lapply(Datos, is.numeric))]
56 data_new
57
58 str(data_new)
59
60 data <- t(scale(t(data_new), center=T, scale=T))
61 data1 <- melt(data)
62
63 head(data1)
64
65 colnames(data1) <- c("Signature", "Cluster", "value")
66
67 write.csv(data1, "~/pDCSignatureNorm.csv", row.names=TRUE)
68
69 data1<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↪
    ↪ pDCSignatureNorm")
70
71 ggplot(data1, aes(x = Cluster, y = Signature, fill = value)) +
72   geom_tile() +
73   labs(title = "pDC Signatures comparison",
74         x = "Cluster",
75         y = "Signature")+
76   guides(fill = guide_colourbar(title = "Expression level"))+
77   scale_fill_gradient2(high = "brown2", mid = "lemonchiffon", low ↪
    ↪ = "royalblue") +
78   theme(axis.text=element_text(size=16, face="bold"),
79         axis.title=element_text(size=13,face="bold"), title = ↪
    ↪ element_text(size=18,face="bold"))

```

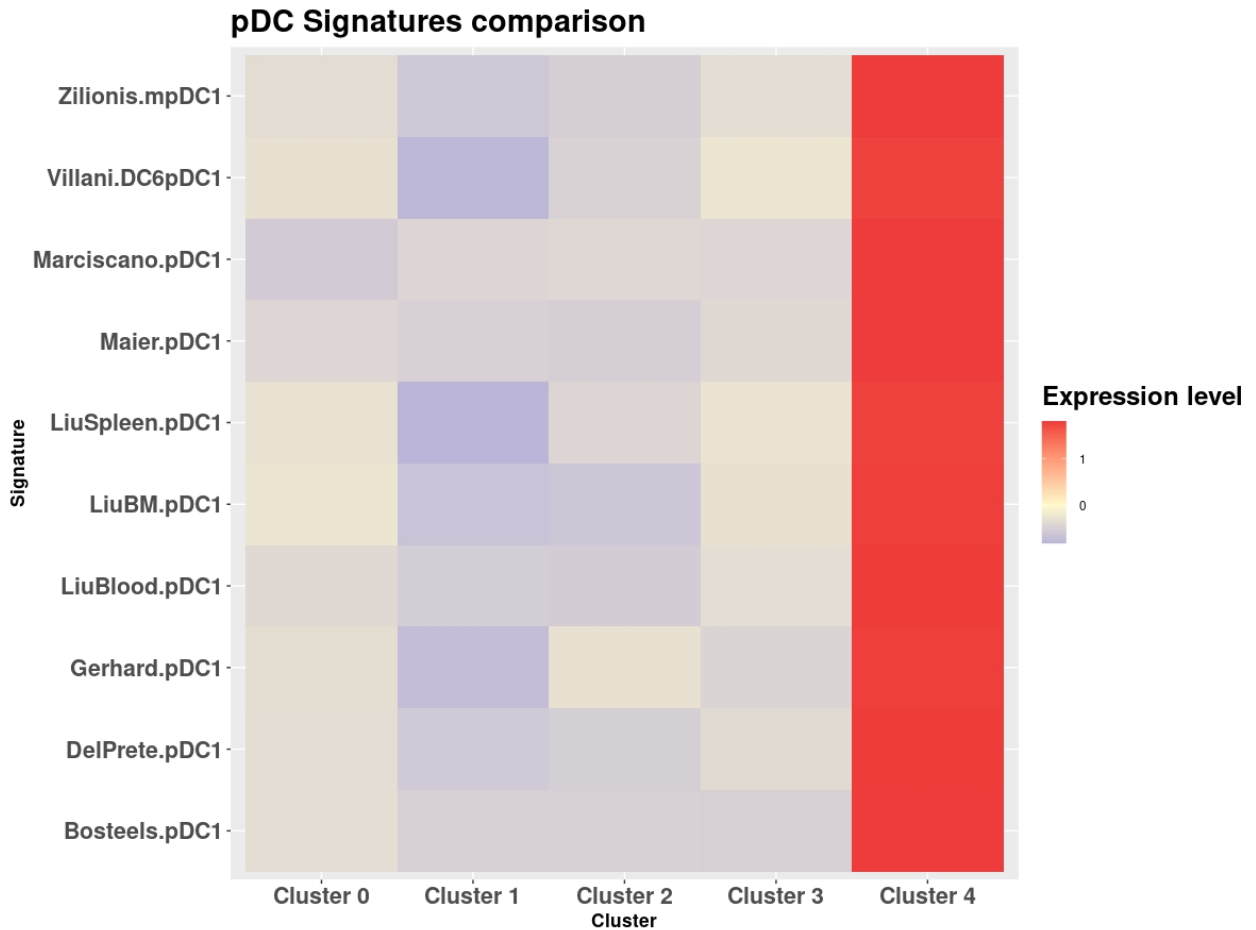



Figure C.23: Heatmap pDC Signatures

Código C.17: R studio code extract – Dendritic cell signature analysis:
MoDC

```

1 #-- MoDC --#
2
3 #Load data
4 data_MoDC<-read.xlsx("DCSignatures.xlsx", sheet = "MoDC")
5
6 Marmonti.moDC <- data_MoDC[,2]
7 Gerhard.cDC2MoDC <- data_MoDC[,3]
8 Marciscano.MoDC <- data_MoDC[,4]
9 Backer.MoDC <- data_MoDC[,5]
10
11 #Top most repeated genes
12 data_MoDC_Top<- c(Marmonti.moDC, Gerhard.cDC2MoDC, Marciscano.MoDC↵
↵ , Backer.MoDC)
13 sort(table(data_MoDC_Top),decreasing=TRUE)[1:10] #at least ↵
↵ repeated 2 times
14

```

```

15 #Gene list
16 Marmonti.moDC <- list(Marmonti.moDC)
17 Gerhard.cDC2MoDC <- list(Gerhard.cDC2MoDC)
18 Marciscano.MoDC <- list(Marciscano.MoDC)
19 Backer.MoDC <- list(Backer.MoDC)
20
21 #Calculate expression level
22 DefaultAssay(object = samplesIntegFltDC) <- "SCT"
23 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
  ↪ features = Marmonti.moDC, name = 'Marmonti.moDC')
24 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
  ↪ features = Gerhard.cDC2MoDC, name = 'Gerhard.cDC2MoDC')
25 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
  ↪ features = Marciscano.MoDC, name = 'Marciscano.MoDC')
26 samplesIntegFltDC <- AddModuleScore(object = samplesIntegFltDC, <-
  ↪ features = Backer.MoDC, name = 'Backer.MoDC')
27
28 #Save in new assay
29 samplesIntegFltDC[['cdmodule']] <- CreateAssayObject(data = t(x = <-
  ↪ FetchData(object = samplesIntegFltDC, vars = c("Marmonti.<-
  ↪ moDC1", "Gerhard.cDC2MoDC1", "Marciscano.MoDC1", "Backer.<-
  ↪ MoDC1"))))
30 combined_averages <- AverageExpression(samplesIntegFltDC, return.<-
  ↪ seurat = FALSE, features = c("Marmonti.moDC1", "Gerhard.<-
  ↪ cDC2MoDC1", "Marciscano.MoDC1", "Backer.MoDC1"), group.by = <-
  ↪ 'integrated_snn_res.0.2', assays = 'cdmodule')
31
32 write.csv(combined_averages, "~/MoDCSignature.csv", row.names=TRUE)
33
34 #Generate normalized data
35 Datos<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "<-
  ↪ MoDCSignature")
36
37 data_new <- Datos[, unlist(lapply(Datos, is.numeric))]
38 data_new
39
40 str(data_new)
41
42 data <- t(scale(t(data_new), center=T, scale=T))
43 data1 <- melt(data)
44
45 head(data1)
46
47 colnames(data1) <- c("Signature", "Cluster", "value")
48

```

```

49 write.csv(data1, "~/MoDCSignatureNorm.csv", row.names=TRUE)
50
51 data1<-read.xlsx("~/DCSignatureExpression.xlsx", sheet = "↔
↪ MoDCSignatureNorm")
52
53 ggplot(data1, aes(x = Cluster, y = Signature, fill = value)) +
54   geom_tile() +
55   labs(title = "MoDC Signatures comparison",
56        x = "Cluster",
57        y = "Signature")+
58   guides(fill = guide_colourbar(title = "Expression level"))+
59   scale_fill_gradient2(high = "brown2", mid = "lemonchiffon", low ↪
↪ = "royalblue") +
60   theme(axis.text=element_text(size=16, face="bold"),
61         axis.title=element_text(size=13,face="bold"), title = ↪
↪ element_text(size=18,face="bold"))

```

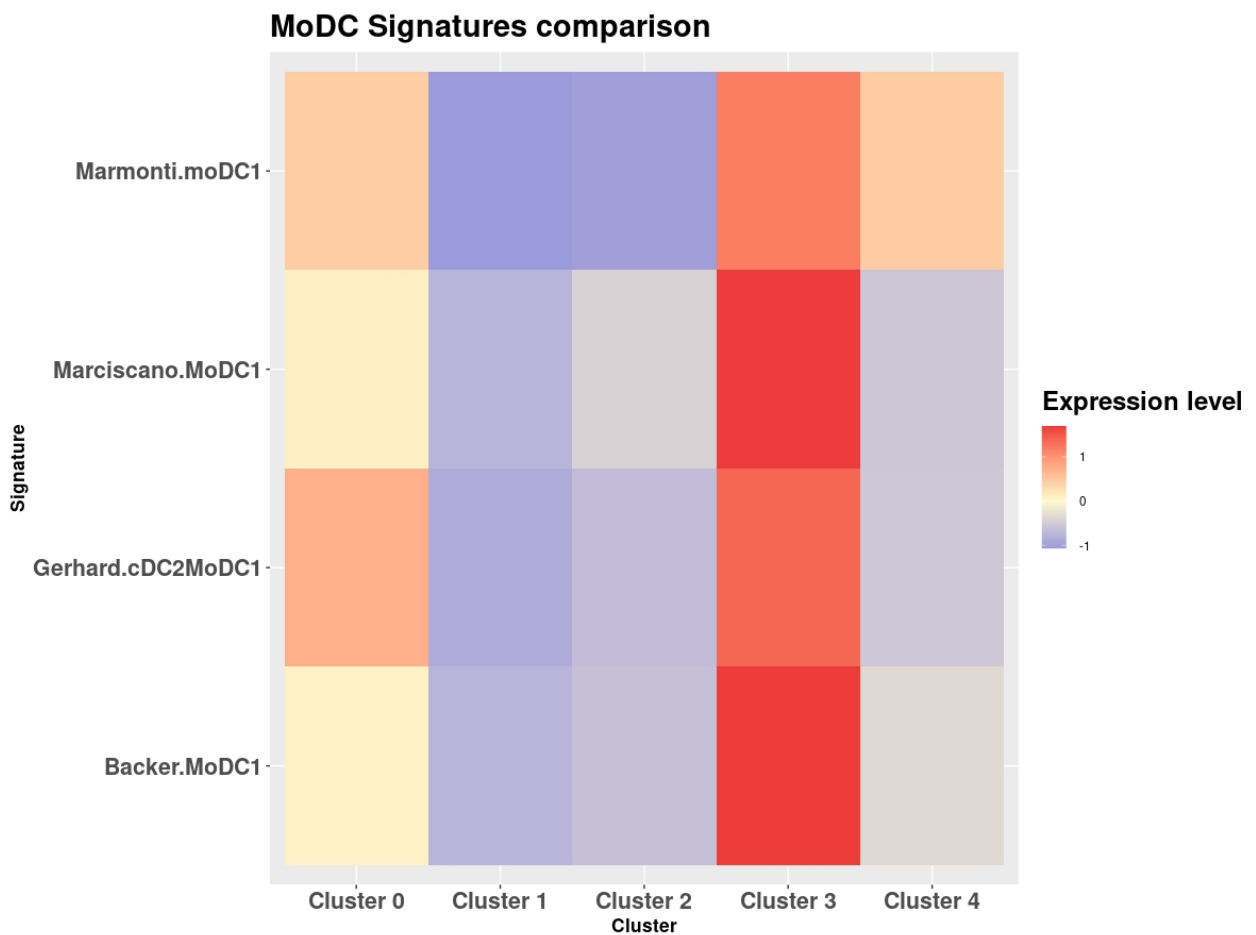


Figure C.24: Heatmap MoDC Signatures

Código C.18: R studio code extract – Dendritic cell subpopulation division

```

1 #-- New classification --#
2 samplesIntegFltDC <- RenameIdents(object = samplesIntegFltDC , 0 = ↵
  ↵ cDC2, 1 = DC3, 2 = cDC1, 3 = MoDC, 4 = pDC)
3
4 DimPlot(samplesIntegFltDC , reduction = "umap" , label = FALSE, pt.↵
  ↵ size = 1.5, cols = my_colors , label.size = 6) + ggtitle('↵
  ↵ Dendritic cells subpopulations')
```

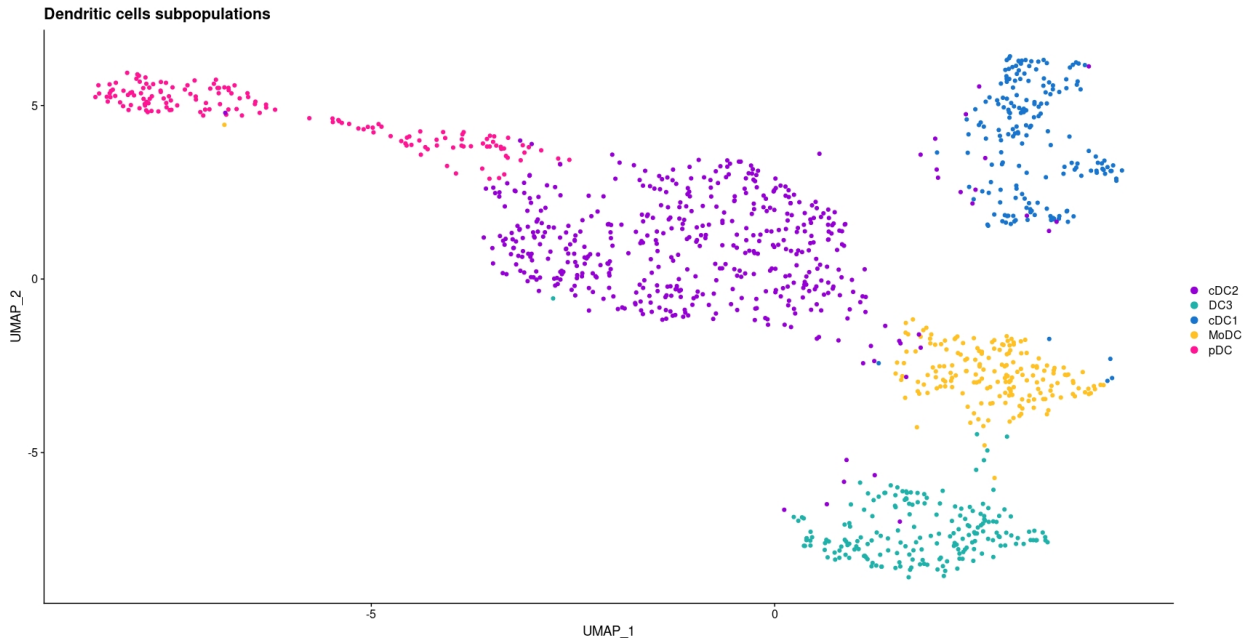


Figure C.25: Dendritic cells subpopulations

Código C.19: R studio code extract – Dendritic cells Barplot

```

1
2 samplesIntegFltDC[["Subpopulation"]] <- samplesIntegFltDC@meta.↵
  ↵ data$integrated_snn_res.0.2
3 dittoBarPlot(object = samplesIntegFltDC , scale = "percent" , var = ↵
  ↵ "Subpopulation" , group.by = "Treatment" , color.panel = my_↵
  ↵ colors , x.labels = c("Untreated" , "Treated") , xlab = "↵
  ↵ Condition" , main = "")
```

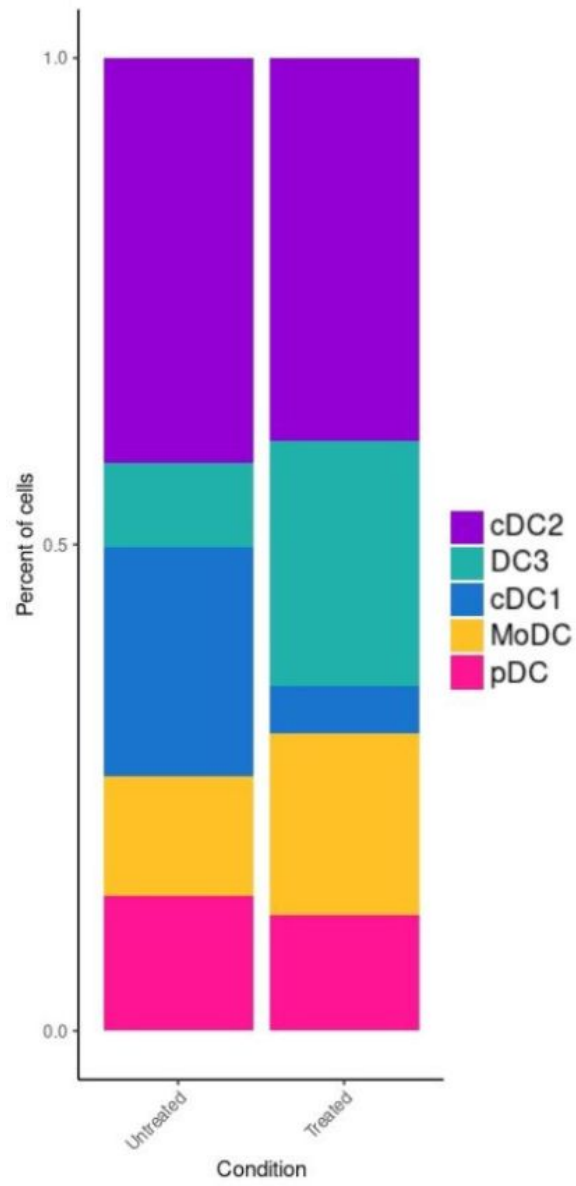


Figure C.26: Barplot of Dendritic cells subpopulations percent by Conditions

Annex D. Monocyte/Macrophages mice analysis code

Código D.1: R studio code extract – Quality Control Monocyte/
Macrophages Part 1

```
1 #----- Monocyte/Macrophages analysis -----#
2
3 #- Separate cell by sample -#
4 samplesIntegrated_MoMa.list <- SplitObject(samplesIntegrated_MoMa, ←
  ↪ split.by = "orig.ident")
5 Untreated1 <- samplesIntegrated_MoMa.list [[1]]
6 Untreated2 <- samplesIntegrated_MoMa.list [[2]]
7 Treated1 <- samplesIntegrated_MoMa.list [[3]]
8 Treated2 <- samplesIntegrated_MoMa.list [[4]]
9
10 #- Check nFeature distribution with Histograms -#
11 Untreated1 <- FetchData(Untreated1, vars=c("nFeature_RNA"))
12 Untreated1 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins←
  ↪ =100) + geom_vline(xintercept = 2000) + ggtitle('Untreated1' ←
  ↪ )
13
14 Untreated2 <- FetchData(Untreated2, vars=c("nFeature_RNA"))
15 Untreated2 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins←
  ↪ =100) + geom_vline(xintercept = 2000) + ggtitle('Untreated2' ←
  ↪ )
16
17 Treated1 <- FetchData(Treated1, vars=c("nFeature_RNA"))
18 Treated1 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins←
  ↪ =100) + geom_vline(xintercept = 2000) + ggtitle('Treated1')
19
20 Treated2 <- FetchData(Treated2, vars=c("nFeature_RNA"))
21 Treated2 %>% ggplot() + geom_histogram(aes(x=nFeature_RNA), bins←
  ↪ =100) + geom_vline(xintercept = 2000) + ggtitle('Treated2')
```

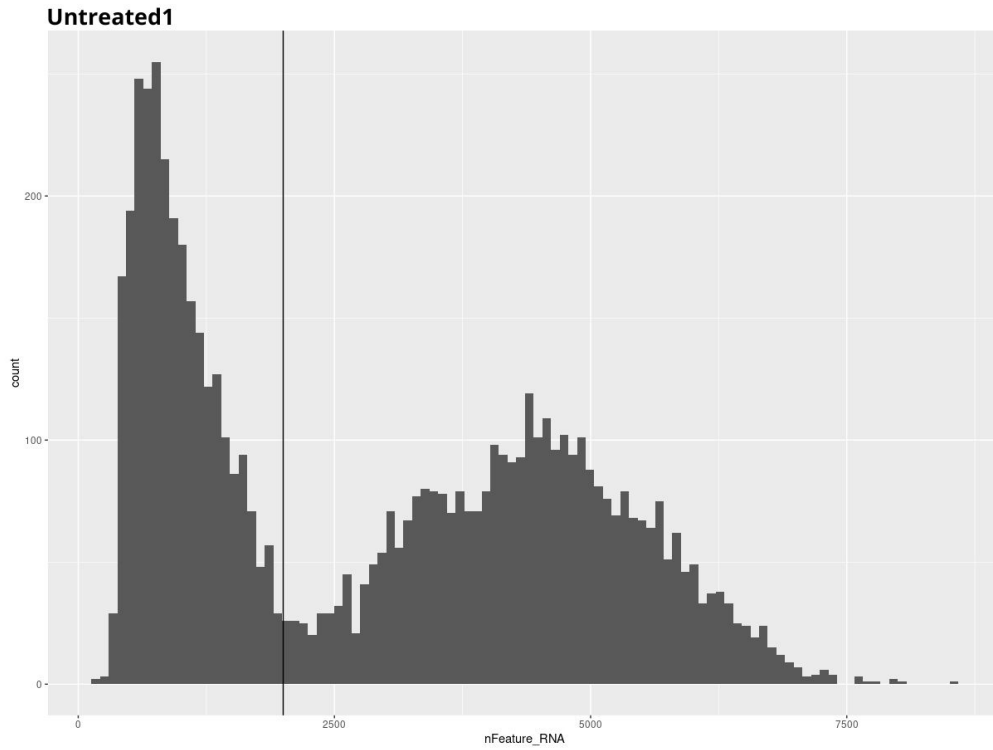


Figure D.1: Histogram of the number of features expressed vs number of cells for Monocyte/Macrophages Untreated sample 1

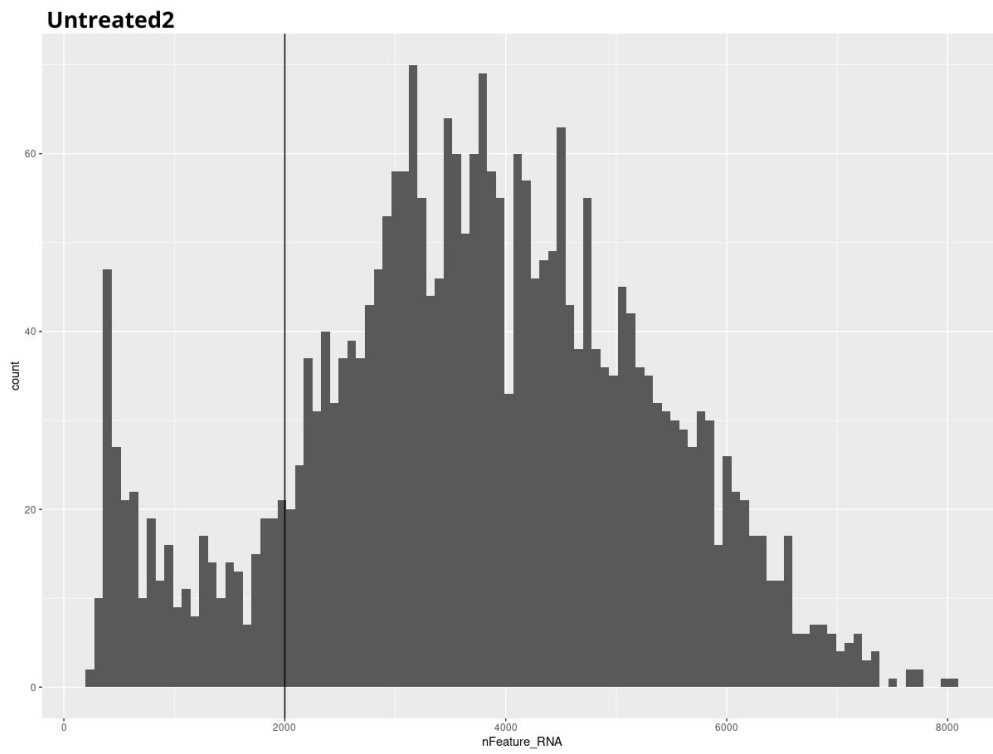


Figure D.2: Histogram of the number of features expressed vs number of cells for Monocyte/Macrophages Untreated sample 2

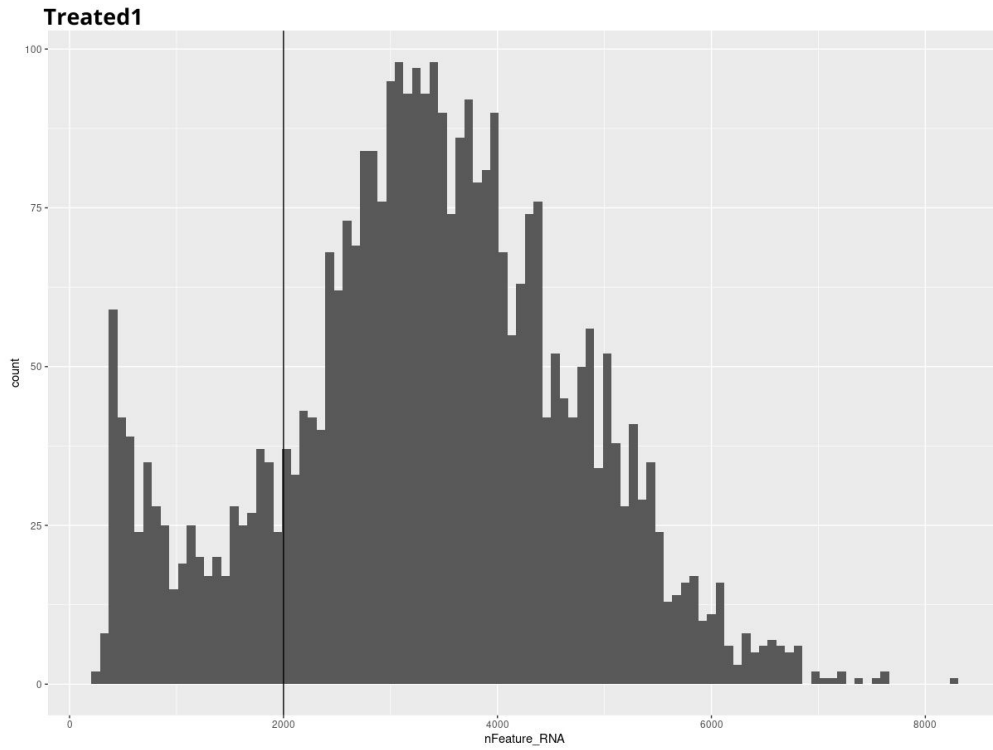


Figure D.3: Histogram of the number of features expressed vs number of cells for Monocyte/Macrophages Treated sample 1

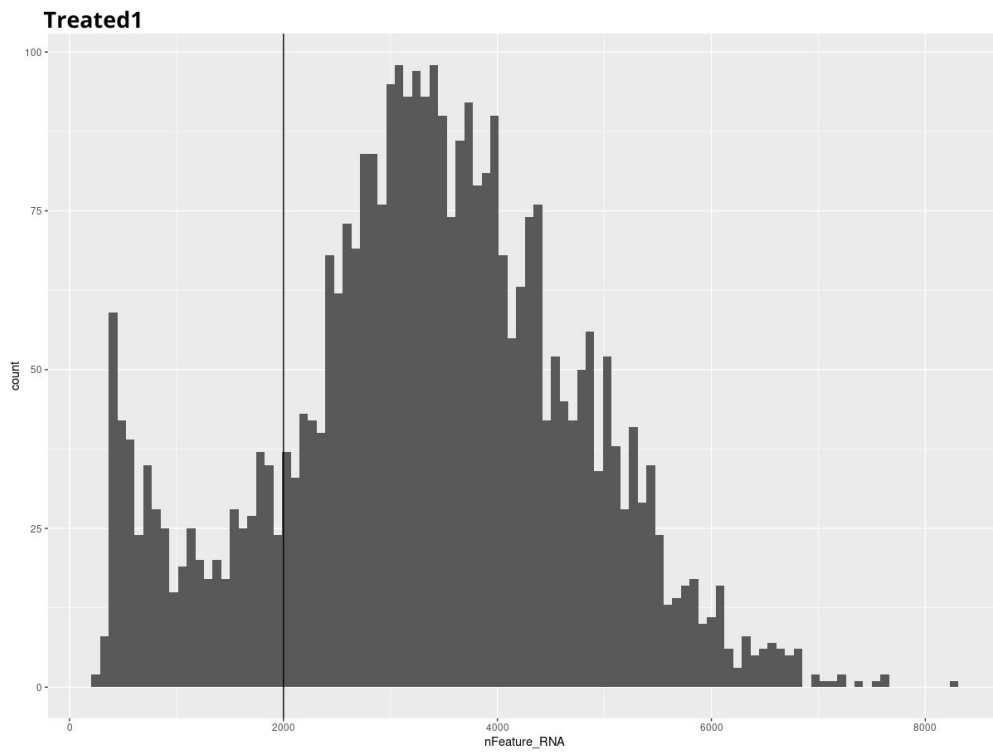


Figure D.4: Histogram of the number of features expressed vs number of cells for Monocyte/Macrophages Treated sample 2

Código D.2: R studio code extract – Quality Control Monocyte/
Macrophages Part 2

```
1 #- Quality Control with Scatter plots -#
2 Untreated1 <- samplesIntegrated_MoMa.list [[1]]
3 Untreated2 <- samplesIntegrated_MoMa.list [[2]]
4 Treated1 <- samplesIntegrated_MoMa.list [[3]]
5 Treated2 <- samplesIntegrated_MoMa.list [[4]]
6
7 #- Untreated 1 -#
8 #Logarithmic data
9 logMT <- FetchData(Untreated1, vars=c("percent.mt"))
10 nFeature <- FetchData(Untreated1, vars=c("nFeature_RNA"))
11
12 Untreated1[["logMT"]] <- log10(logMT)
13 Untreated1[["logFeature"]] <- log10(nFeature)
14
15 #Scatter plots
16 p1 <- FeatureScatter(Untreated1, feature1="nFeature_RNA", feature2=<-
  ↪="percent.mt", group.by = "orig.ident") + geom_hline(<-
  ↪ yintercept=10) + ggtitle('Untreated1') + geom_vline(<-
  ↪ xintercept = 2000)
17 p2 <- FeatureScatter(Untreated1, feature1="logFeature", feature2="<-
  ↪ logMT", group.by = "orig.ident") + geom_hline(yintercept=1) <-
  ↪ + ggtitle('Untreated1') + geom_vline(xintercept = 3.301)
18 p1+p2
19
20 #Create subset with new data
21 Untreated1 <- subset(Untreated1, subset = nFeature_RNA > 2000)
22
23
24 #- Untreated 2 -#
25 #Logarithmic data
26 logMT <- FetchData(Untreated2, vars=c("percent.mt"))
27 nFeature <- FetchData(Untreated2, vars=c("nFeature_RNA"))
28
29 Untreated2[["logMT"]] <- log10(logMT)
30 Untreated2[["logFeature"]] <- log10(nFeature)
31
32 #Initial Scatter plots
33 p1 <- FeatureScatter(Untreated2, feature1="nFeature_RNA", feature2=<-
  ↪="percent.mt", group.by = "orig.ident") + geom_hline(<-
  ↪ yintercept=10) + ggtitle('Untreated2') + geom_vline(<-
  ↪ xintercept = 2000)
34 p2 <- FeatureScatter(Untreated2, feature1="logFeature", feature2="<-
  ↪ logMT", group.by = "orig.ident")+ geom_hline(yintercept=1) <-
```

```

    ↪ ggtitle('Untreated2') + geom_vline(xintercept = 3.301)
35 p1+p2
36
37 #Create subset with new data
38 Untreated2 <- subset(Untreated2, subset = nFeature_RNA > 2000)
39
40
41 #- Treated 1 -#
42 #Logarithmic data
43 logMT <- FetchData(Treated1, vars=c("percent_mt"))
44 nFeature <- FetchData(Treated1, vars=c("nFeature_RNA"))
45
46 Treated1[["logMT"]] <- log10(logMT)
47 Treated1[["logFeature"]] <- log10(nFeature)
48
49 #Scatter plots
50 p1 <- FeatureScatter(Treated1, feature1="nFeature_RNA", feature2="↪
    ↪ percent_mt", group.by = "orig.ident") + geom_hline(↪
    ↪ yintercept=7) + ggtitle('Treated1') + geom_vline(xintercept ↪
    ↪ = 2000)
51 p2 <- FeatureScatter(Treated1, feature1="logFeature", feature2="↪
    ↪ logMT", group.by = "orig.ident") + geom_hline(yintercept↪
    ↪ =0.85) + ggtitle('Treated1') + geom_vline(xintercept = ↪
    ↪ 3.301)
52 p1+p2
53
54 #Create subset with new data
55 Treated1 <- subset(Treated1, subset = nFeature_RNA > 2000)
56
57
58 #- Treated 2 -#
59 #Log graphics
60 logMT <- FetchData(Treated2, vars=c("percent_mt"))
61 nFeature <- FetchData(Treated2, vars=c("nFeature_RNA"))
62
63 Treated2[["logMT"]] <- log10(logMT)
64 Treated2[["logFeature"]] <- log10(nFeature)
65
66 #Scatter plots
67 p1 <- FeatureScatter(Treated2, feature1="nFeature_RNA", feature2="↪
    ↪ percent_mt", group.by = "orig.ident") + geom_hline(↪
    ↪ yintercept=8) + ggtitle('Treated2') + geom_vline(xintercept ↪
    ↪ = 2000)
68 p2 <- FeatureScatter(Treated2, feature1="logFeature", feature2="↪
    ↪ logMT", group.by = "orig.ident") + geom_hline(yintercept↪

```

```

↪ =0.9) + ggtitle('Treated2') + geom_vline(xintercept = 3.301)
69 p1+p2
70
71 #Create subset with new data
72 Treated2 <- subset(Treated2, subset = nFeature_RNA > 2000)

```

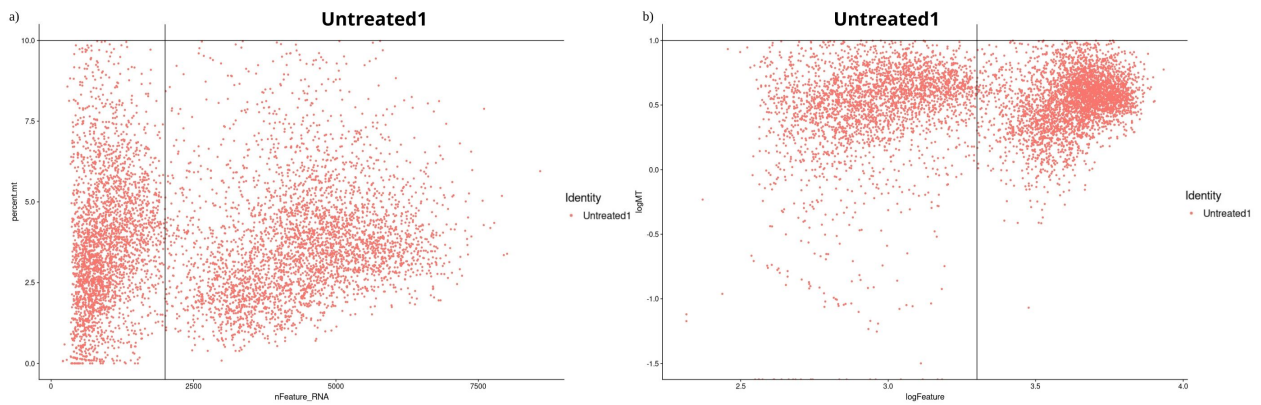


Figure D.5: Scatter plot of Monocyte/Macrophages Untreated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

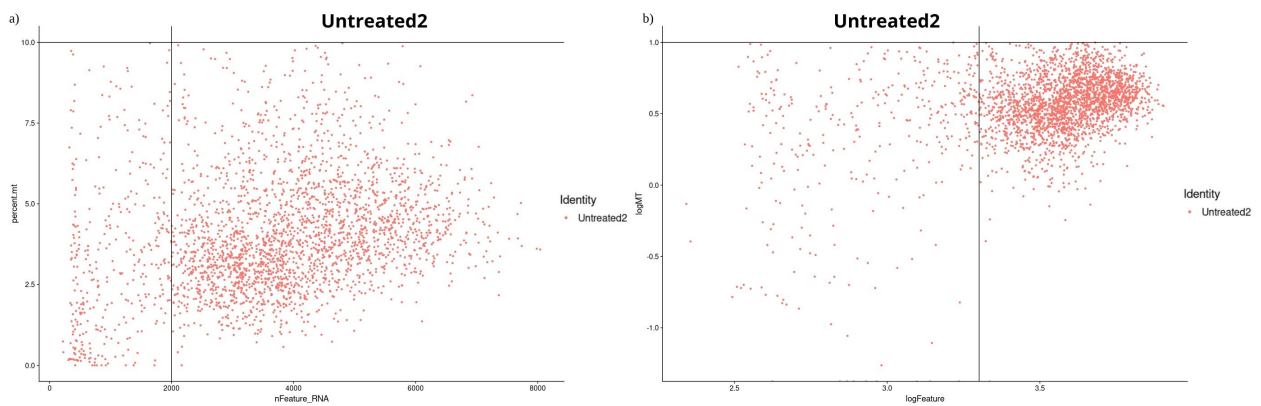


Figure D.6: Scatter plot of Monocyte/Macrophages Untreated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

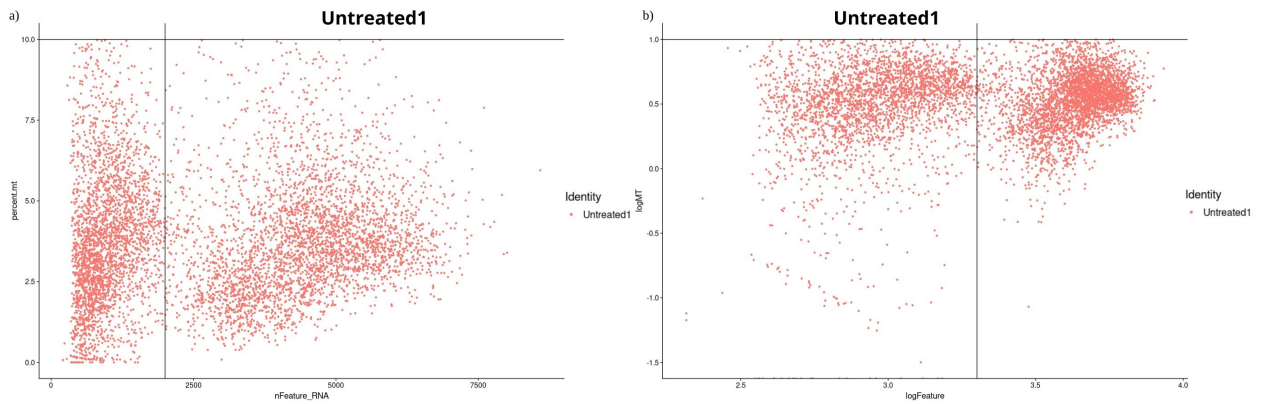


Figure D.7: Scatter plot of Monocyte/Macrophages Treated sample 1 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

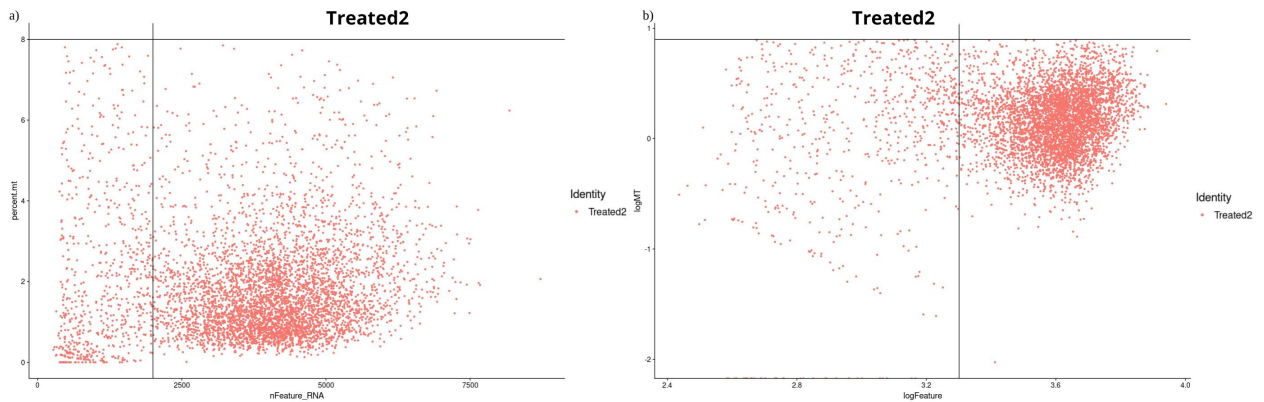


Figure D.8: Scatter plot of Monocyte/Macrophages Treated sample 2 of mitochondrial gene content versus number of features expressed. a) Scatter plot with raw data values. b) Scatter plot with logarithmic values.

Código D.3: R studio code extract – Monocyte/Macrophages Integration

```

1 #--- Integration ---#
2
3 samplesMoMa.list <- list(Untreated1, Untreated2, Treated1, ←
  ↪ Treated2)
4
5 #- Normalice samples -#
6 samplesMoMa.list <- lapply(X= samplesMoMa.list , FUN = function(x){
7   x <- SCTransform(object = x, vars.to.regress = "percent.mt", ←
  ↪ verbose = FALSE, variable.features.n = 5000)})
8
9 #- Select integration features and prepare for integration -#
10 samplesMoMa.features <- SelectIntegrationFeatures(object.list = ←
  ↪ samplesMoMa.list )

```

```

11 samplesMoMa.list <- PrepSCTIntegration(object.list = samplesMoMa.<-
    ↪ list , anchor.features = samplesMoMa.features)
12
13 #- Select integration anchors -#
14 samplesMoMa.anchors <- FindIntegrationAnchors(object.list = <-
    ↪ samplesMoMa.list , normalization.method = "SCT", anchor.<-
    ↪ features = samplesMoMa.features)
15 save(samplesMoMa.anchors , file=file.path("~/", "samplesMoMa.anchors<-
    ↪ .rdata"))
16
17 #- Integrate -#
18 samplesIntegFltMoMa <- IntegrateData(anchorset = samplesMoMa.<-
    ↪ anchors , normalization.method = "SCT")
19 saveRDS(samplesIntegFltMoMa , "samplesIntegFltMoMa.rds")

```

Código D.4: R studio code extract – Monocyte/Macrophages linear reduction

```

1 #- PCA reduction -#
2 samplesIntegFltMoMa<- RunPCA(samplesIntegFltMoMa , npcs = 50, <-
    ↪ verbose=F)
3
4 #Elbow plot
5 ElbowPlot(samplesIntegFltMoMa , ndims = 50, reduction = "pca") + <-
    ↪ ggtitle("Elbow plot Integrated MoMa data")

```

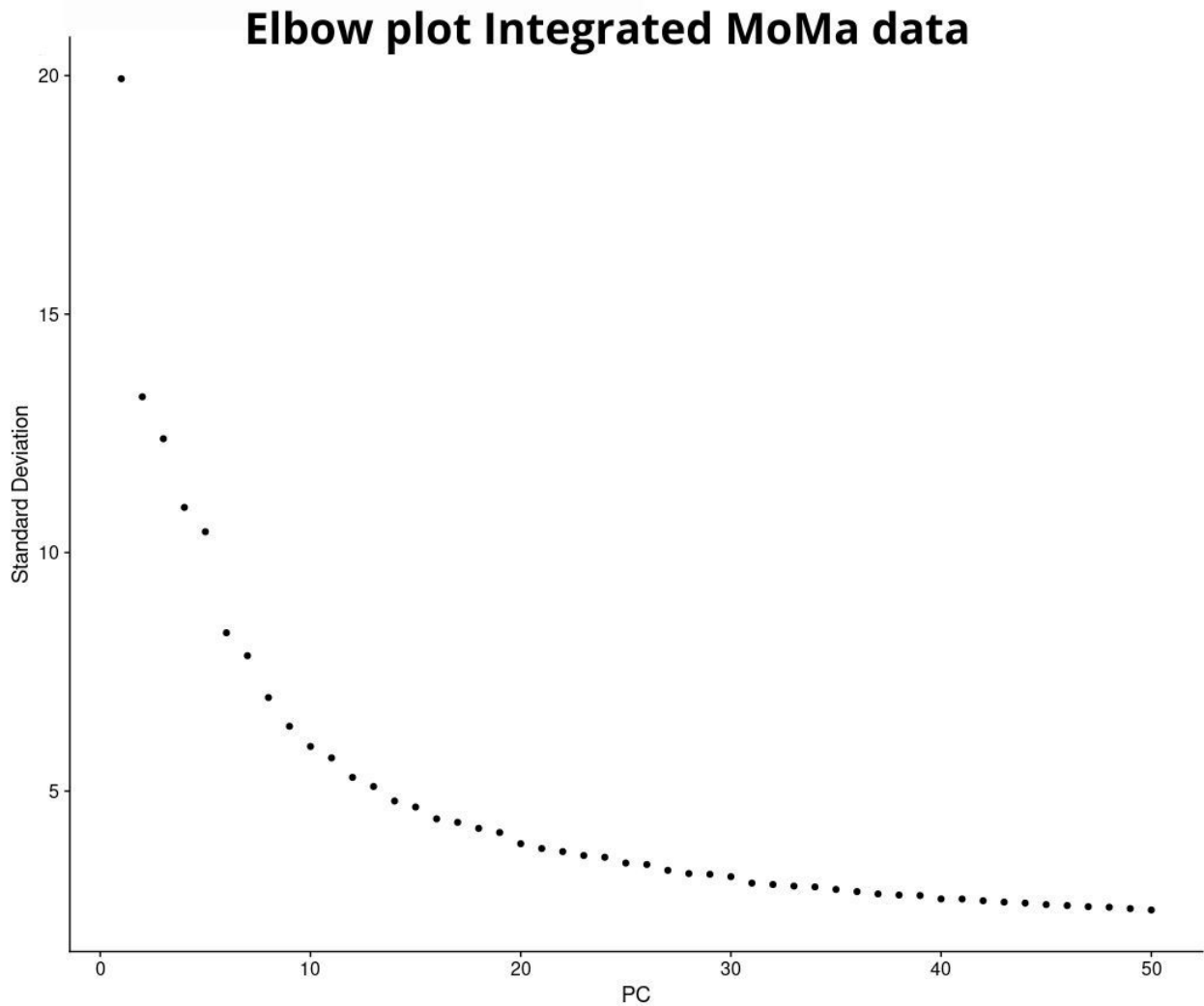


Figure D.9: Elbow plot Monocyte/Macrophages

Código D.5: R studio code extract – Monocyte/Macrophages non-linear reduction

```

1 #- UMAP Reduction -#
2 samplesIntegFltMoMa <- RunUMAP(samplesIntegFltMoMa, dims=1:40, ↵
  ↵ verbose=F)
3
4 #UMAP by Condition
5 DimPlot(samplesIntegFltMoMa, reduction = "umap", group.by = '↵
  ↵ Treatment', pt.size = 1.5) + ggtitle("Integrated MoMa UMAP ↵
  ↵ divided by Condition") + scale_color_manual(labels = c("↵
  ↵ Untreated", "Treated"), values = c("#F8766D", "#00bfc4"))
6
7 #UMAP by Sample
8 DimPlot(samplesIntegFltMoMa, reduction = "umap", group.by = 'orig.↵
  ↵ ident', pt.size = 1.5) + ggtitle("Integrated MoMa UMAP ↵

```

```
↪ divided by Sample") + scale_color_manual(labels = c("↪  
↪ Untreated1", "Untreated2", "Treated1", "Treated2"), values = ↪  
↪ my_colors)
```

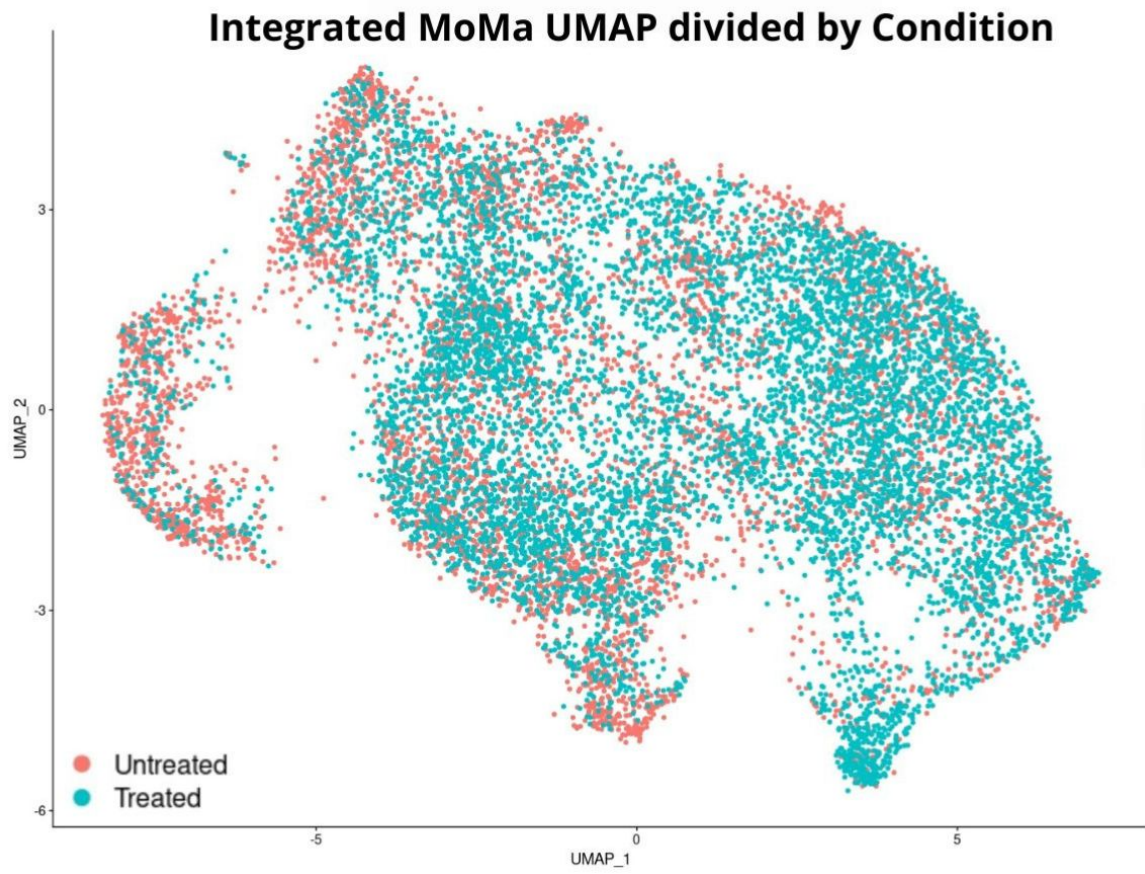


Figure D.10: UMAP of Monocyte/Macrophages divided by Condition

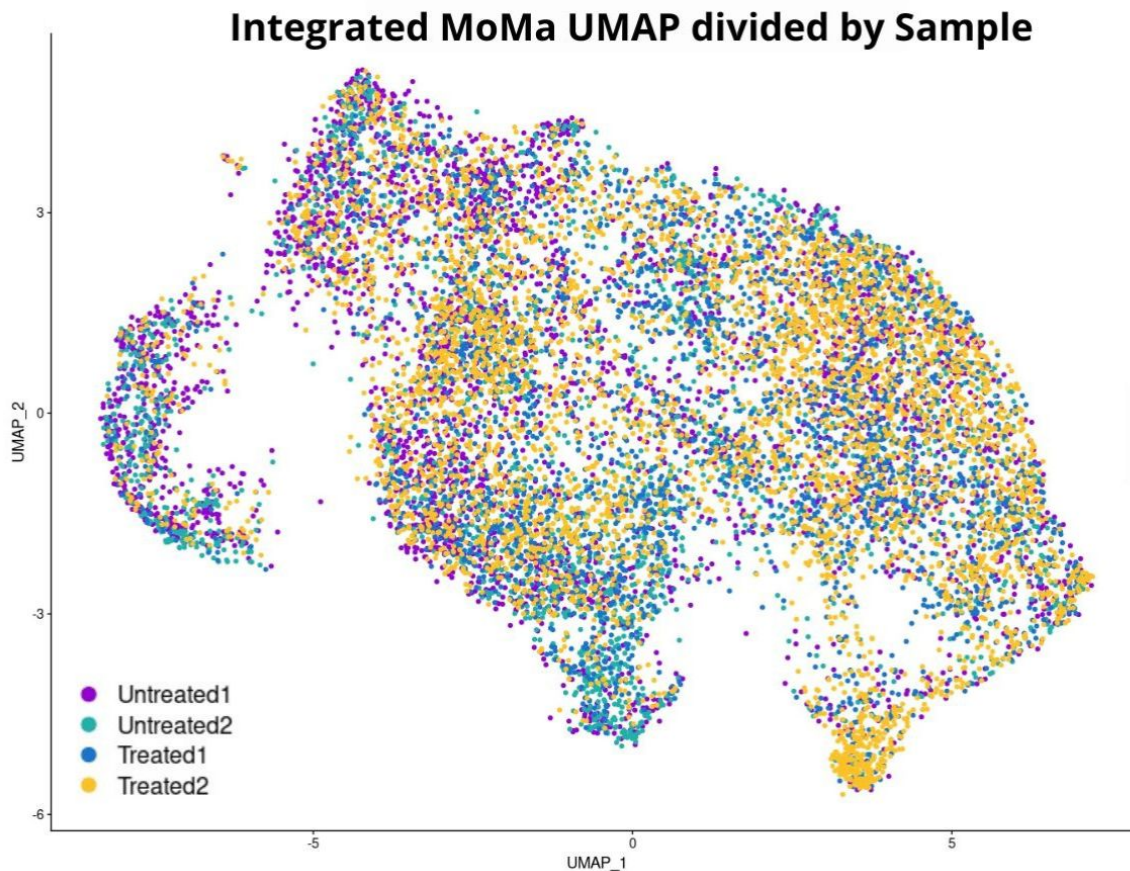


Figure D.11: UMAP of Monocyte/Macrophages divided by Sample

Código D.6: R studio code extract – Monocyte/Macrophages clustering

```

1 #--- Clustering ---#
2 DefaultAssay(object = samplesIntegFltMoMa) <- "integrated"
3
4 samplesIntegFltMoMa <- FindNeighbors(samplesIntegFltMoMa, dims = <-
  ↪ 1:50, verbose = FALSE)
5
6 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, <-
  ↪ resolution = 0)
7 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.<-
  ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0 Monocyte<-
  ↪ /Macrophages')
8
9 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, <-
  ↪ resolution = 0.1)
10 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.<-
  ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.1 <-
  ↪ Monocyte/Macrophages')
11

```



```

12 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.2)
13 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.2 ←
    ↪ Monocyte/Macrophages')
14
15 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.3)
16 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.3 ←
    ↪ Monocyte/Macrophages')
17
18 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.4)
19 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.4 ←
    ↪ Monocyte/Macrophages')
20
21 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.5)
22 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.5 ←
    ↪ Monocyte/Macrophages')
23
24 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.6)
25 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.6 ←
    ↪ Monocyte/Macrophages')
26
27 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.7)
28 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.7 ←
    ↪ Monocyte/Macrophages')
29
30 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.8)
31 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.8 ←
    ↪ Monocyte/Macrophages')
32
33 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ←
    ↪ resolution = 0.9)
34 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪

```

```
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 0.9 ↪
    ↪ Monocyte/Macrophages')
35
36 samplesIntegFltMoMa <- FinMoMalusters(samplesIntegFltMoMa, ↪
    ↪ resolution = 1.0)
37 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = TRUE, pt.↪
    ↪ size = 1, cols = my_colors) + ggtitle('Resolution 1.0 ↪
    ↪ Monocyte/Macrophages')
38
39 clustree(samplesIntegFltMoMa)
```

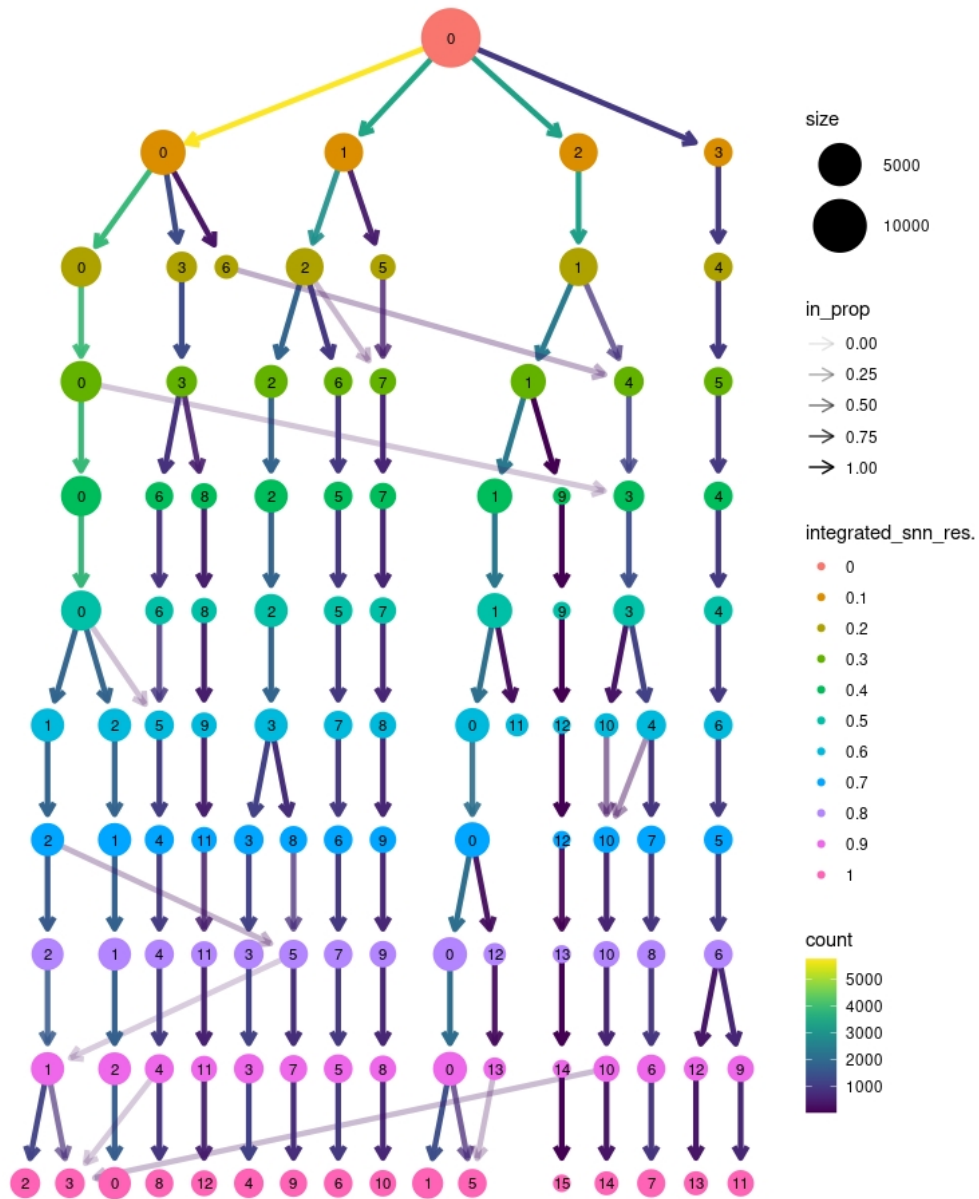


Figure D.12: Monocyte/Macrophages Clustree

Código D.7: R studio code extract – Remosion of contaminant Granulocytes

```

1
2 #-- Remove granulocyte contaminants --#
3 samplesIntegFltMoMa <- FindClusters(samplesIntegFltMoMa, ↵
  ↵ resolution = 0.7)
4 DimPlot(samplesIntegFltMoMaNew, reduction = "umap", label = FALSE, ↵
  ↵ pt.size = 1, cols = my_colors) + ggtitle('Resolution 0.7 ↵
  ↵ Monocyte/Macrophages cells')
5

```

```

6 DefaultAssay(object = samplesIntegFltMoMa) <- "SCT"
7 VlnPlot(samplesIntegFltMoMa, features = "Mmp9", pt.size = 0, cols ←
  ↪ = my_colors)
8
9 ContaminantGranu <- subset(samplesIntegFltMoMa, idents= 12, invert←
  ↪ =FALSE)
10 samplesIntegFltMoMa <- subset(samplesIntegFltMoMa, idents= 12, ←
  ↪ invert=TRUE)
11
12 saveRDS(ContaminantGranu, "ContaminantGranu.rds")
13
14 saveRDS(samplesIntegFltMoMa, "samplesIntegFltMoMaNoCont.rds")
15
16 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = FALSE, pt←
  ↪ .size = 1, cols = my_colors) + ggtitle('Resolution 0.7 ←
  ↪ Monocyte/Macrophages cells')
17
18 DimPlot(ContaminantGranu, reduction = "umap", label = FALSE, pt.←
  ↪ size = 1, cols = "salmon4") + ggtitle('Contaminant ←
  ↪ Granulocytes')
19
20
21 #New Clustree
22 clustree(samplesIntegFltMoMa)

```

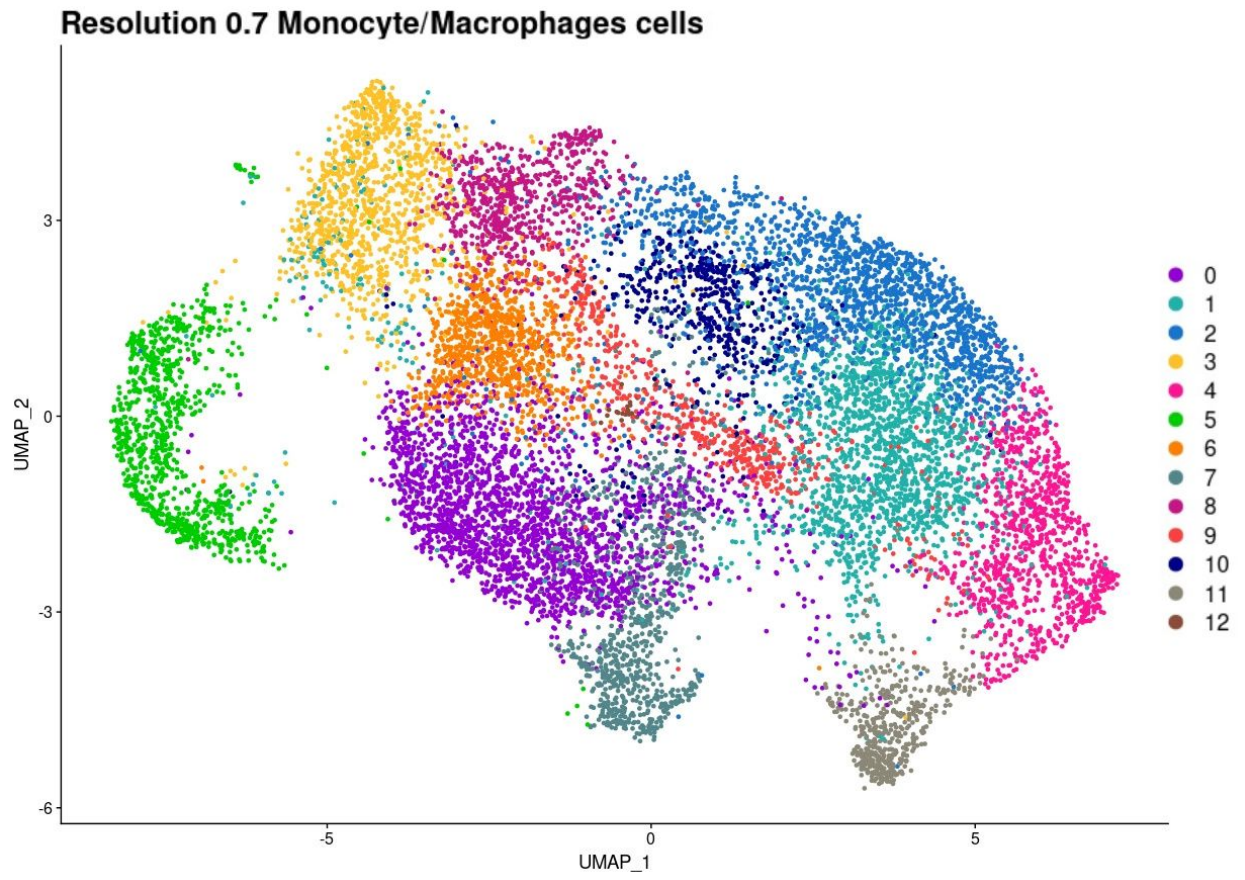


Figure D.13: Integrated Monocyte/Macrophages cells UMAP resolution 0.7

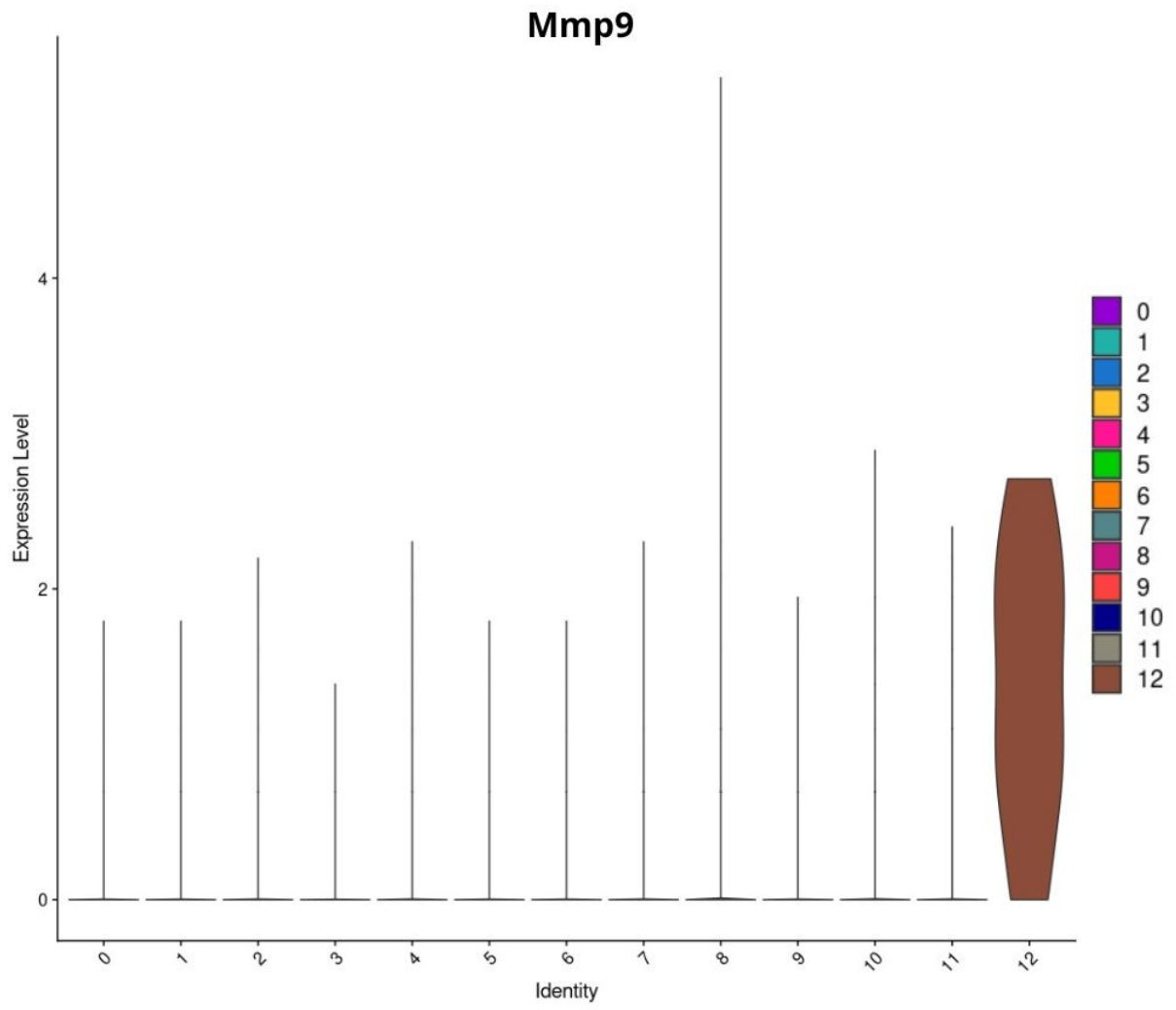


Figure D.14: Mmp9 violin plot

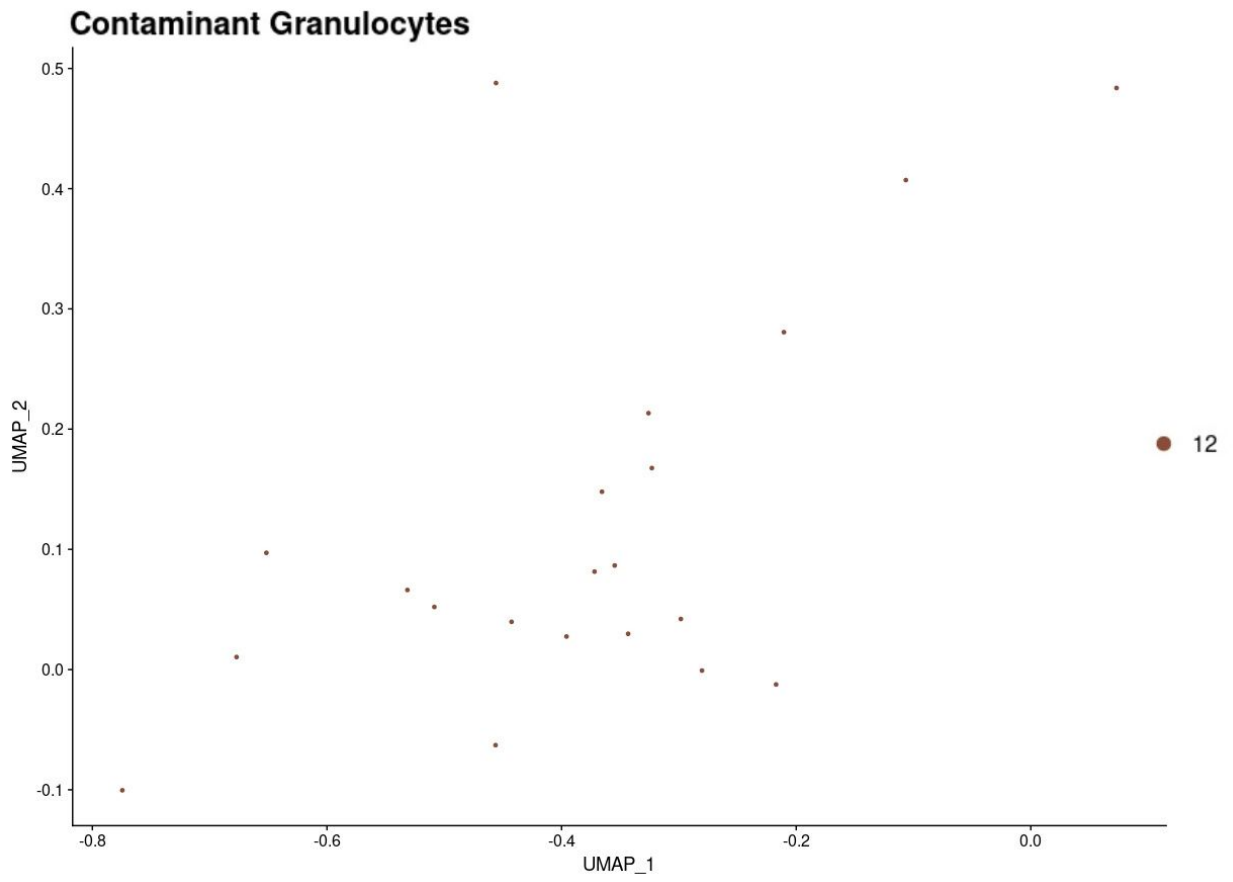


Figure D.15: Contaminant granulocytes clusters

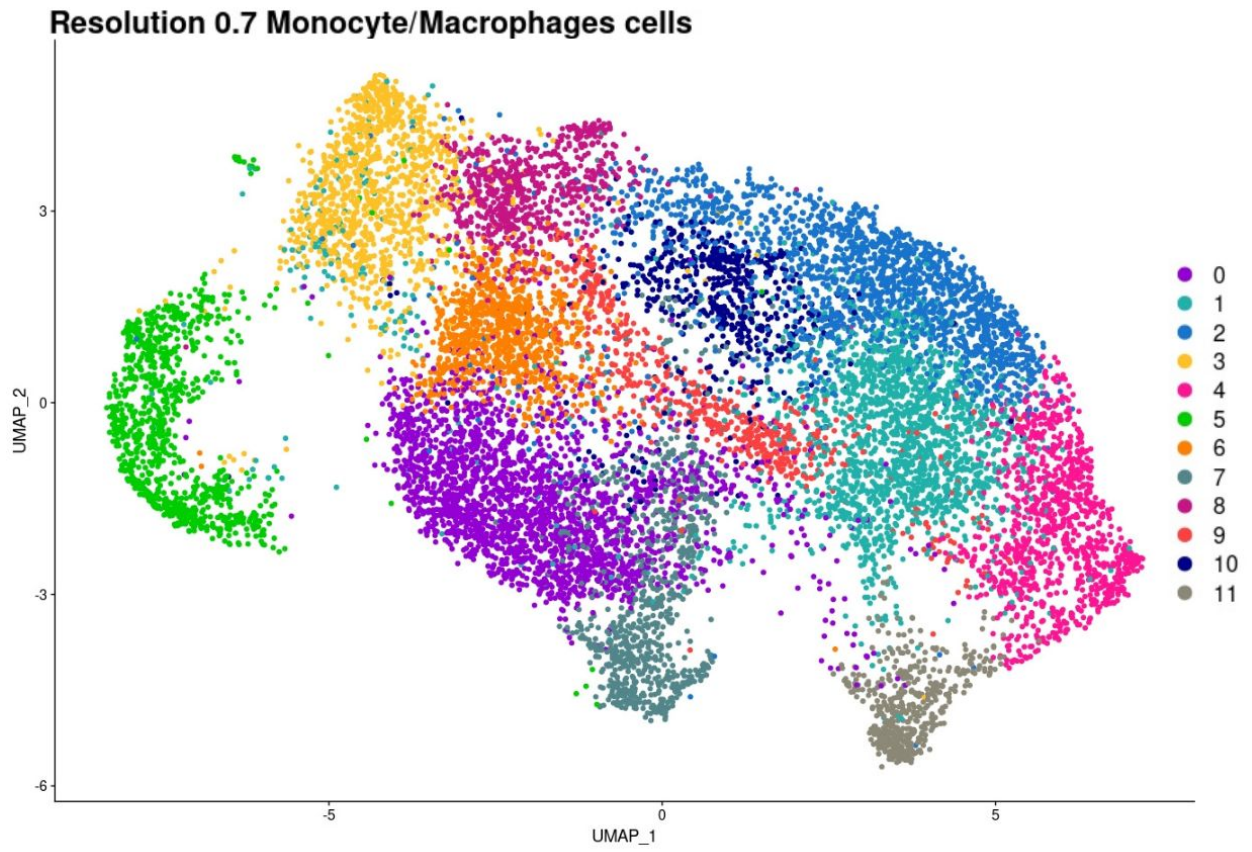


Figure D.16: New integrated Monocyte/Macrophages cells UMAP resolution 0.7 without contaminants

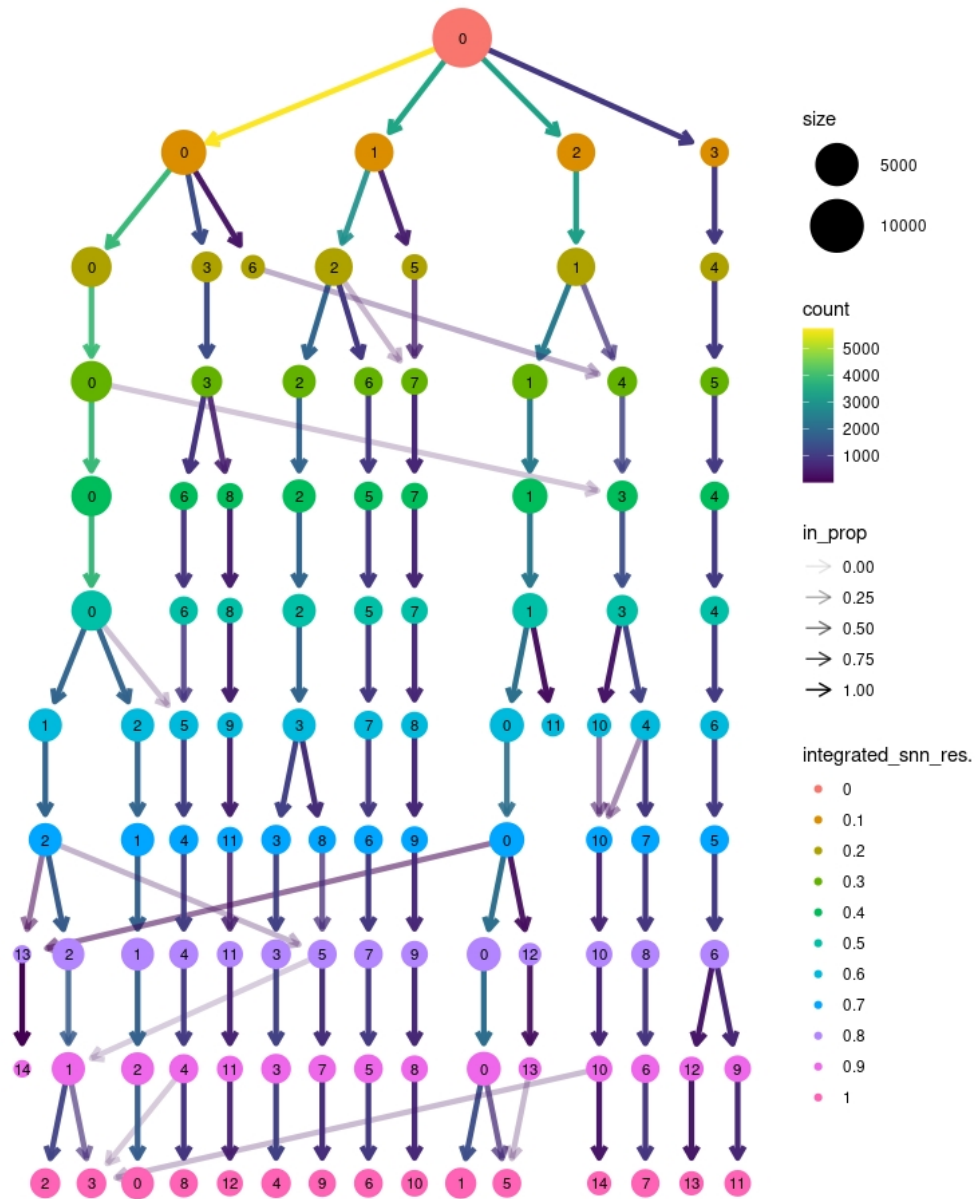


Figure D.17: New Monocyte/Macrophages Clustree without contaminants

Código D.8: R studio code extract – Monocyte/Macrophages density plot

```

1 #--- Density plot ---#
2 DefaultAssay(object = samplesIntegFltMoMa) <- "SCT"
3
4 sampsList <- SplitObject(samplesIntegFltMoMa, split.by = "↔
  ↳ Treatment")
5
6 MoMa_Untreated <- sampsList[[1]]
7 MoMa_Treated <- sampsList[[2]]

```

```

8
9 saveRDS(MoMa_Untreated, "MoMa_Untreated.rds")
10 saveRDS(MoMa_Treated, "MoMa_Treated.rds")
11
12 #- Normal -#
13 p <- DimPlot(samplesIntegFltMoMa, group.by = "Treatment", reduction<-
  ↪ = "umap")
14
15 #Prepare data
16 data <- data.frame(samplesIntegFltMoMa@meta.data)
17 data$UMAP1 <- Embeddings(samplesIntegFltMoMa, "umap")[,1]
18 data$UMAP2 <- Embeddings(samplesIntegFltMoMa, "umap")[,2]
19
20 #Create a data frame that doesn't contain a "sample" column.
21 data_bg <- data[, -(which(colnames(data)=="Sample"))]
22
23 #Plot
24 density_plot <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
25   geom_point(colour="#00000000") +
26   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins<-
  ↪ =25) +
27   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
  ↪ darkolivegreen3", "goldenrod1", "red")) +
28   xlim(c(-10, 10)) + ylim(c(-10, 10)) +
29   #facet_wrap(~Treatment, ncol=3)+
30   theme_classic() + ggtitle("Density") +
31   theme(plot.title = element_text(hjust = 0.5), text = element_↪
  ↪ text(size = 14))
32
33 density_plot
34
35 #- density 2: Untreated -#
36
37 #Prepare data
38 data <- data.frame(MoMa_Untreated@meta.data)
39 data$UMAP1 <- Embeddings(MoMa_Untreated, "umap")[,1]
40 data$UMAP2 <- Embeddings(MoMa_Untreated, "umap")[,2]
41
42 #Create a data frame that doesn't contain a "sample" column.
43 data_bg <- data[, -(which(colnames(data)=="Sample"))]
44
45 #Plot
46 density_plot_2 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
47   geom_point(colour="#00000000") +
48   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins<-

```

```

    ↪ =25) +
49 scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
50 xlim(c(-10, 10)) + ylim(c(-12, 10)) +
51 facet_wrap(~Treatment, ncol=3)+
52 theme_classic() + ggtitle("Untreated Density plot") +
53 theme(plot.title = element_text(hjust = 0.5), text = element_↪
    ↪ text(size = 14))
54
55 density_plot_2
56
57 #- density 3: Treated -#
58
59 #Prepare data
60 data <- data.frame(MoMa_Treated@meta.data)
61 data$UMAP1 <- Embeddings(MoMa_Treated, "umap")[,1]
62 data$UMAP2 <- Embeddings(MoMa_Treated, "umap")[,2]
63
64 #Create a data frame that doesn't contain a "sample" column.
65 data_bg <- data[, -(which(colnames(data)=="Sample"))]
66
67 #Plot
68 density_plot_3 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
69   geom_point(colour="#00000000") +
70   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↪
    ↪ =25) +
71   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
72   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
73   facet_wrap(~Treatment, ncol=3)+
74   theme_classic() + ggtitle("Treated Density plot") +
75   theme(plot.title = element_text(hjust = 0.5), text = element_↪
    ↪ text(size = 14))
76
77 density_plot_3
78
79
80 plot_grid(p, density_plot, density_plot_2, density_plot_3, ncol=2)

```

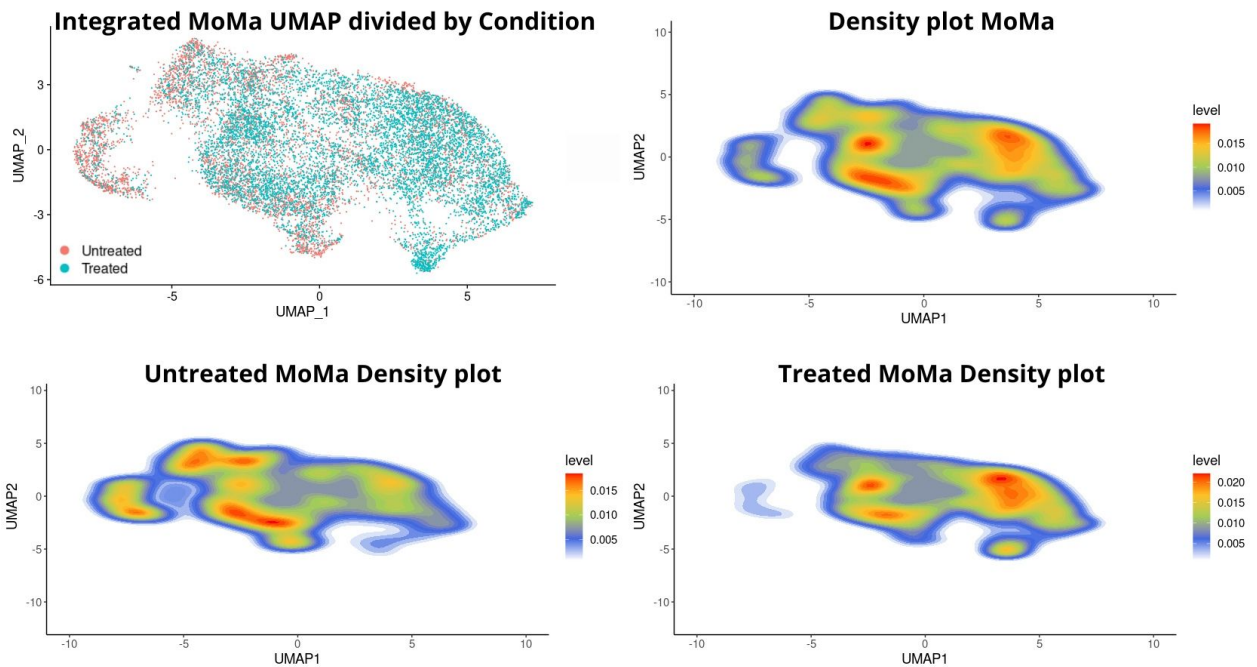


Figure D.18: Monocyte/Macrophages density plots

Código D.9: R studio code extract – Monocyte/Macrophages differential gene expression analysis

```

1 DefaultAssay(object = samplesIntegFltMoMa) <- "integrated"
2
3 samplesIntegFltMoMa <- FindClusters(samplesIntegFltMoMa, ↵
  ↵ resolution = 0.7)
4 samplesIntegFltMoMa.markers <- FindAllMarkers(samplesIntegFltMoMa, ↵
  ↵ only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
5
6 top10Markers <- samplesIntegFltMoMa.markers %>% group_by(cluster) ↵
  ↵ %>% top_n(n=10, wt=avg_log2FC)
7 top_10 <- unique(top10Markers$gene)
8 DotPlot(samplesIntegFltMoMa, features = top_10, cols=c("royalblue" ↵
  ↵ , "brown2")) + RotatedAxis() + ggtitle('Resolution 0.7 ↵
  ↵ markers')
9
10 write.csv(samplesIntegFltMoMa.markers, "~/samplesIntegFltMoMa. ↵
  ↵ markers.csv", row.names=TRUE)

```

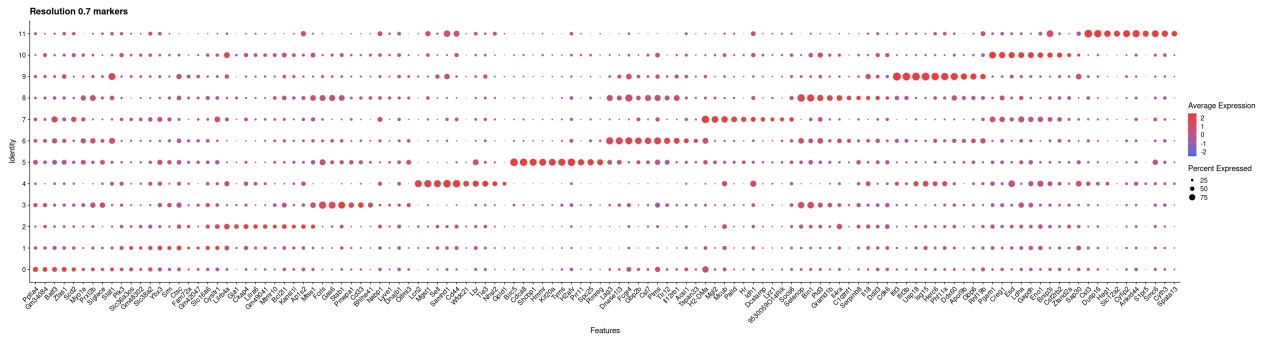


Figure D.19: Monocyte/Macrophages resolution 0.7 Dotplot

Código D.10: R studio code extract – Monocyte/Macrophages gene expression analysis

```

1 DefaultAssay(object = samplesIntegFltMoMa) <- "SCT"
2
3 p1 <- FeaturePlot(samplesIntegFltMoMa, features = 'Csf1r', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
4 p2 <- FeaturePlot(samplesIntegFltMoMa, features = 'Ly6c2', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
5 p3 <- FeaturePlot(samplesIntegFltMoMa, features = 'Spn', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
6 p4 <- FeaturePlot(samplesIntegFltMoMa, features = 'Rps8', min.<-
  ↪ cutoff = 3.2, pt.size = 2, max.cutoff = 10, cols = c("↪
  ↪ royalblue", "brown2"))
7 p5 <- FeaturePlot(samplesIntegFltMoMa, features = 'Ednrb', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
8 p6 <- FeaturePlot(samplesIntegFltMoMa, features = 'Sell', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
9 p7 <- FeaturePlot(samplesIntegFltMoMa, features = 'H2-Aa', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
10 p8 <- FeaturePlot(samplesIntegFltMoMa, features = 'Apoe', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
11 p9 <- FeaturePlot(samplesIntegFltMoMa, features = 'Hmox1', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
12 p10 <- FeaturePlot(samplesIntegFltMoMa, features = 'Fcrls', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
13 p11 <- FeaturePlot(samplesIntegFltMoMa, features = 'Cxcl9', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
14 p12 <- FeaturePlot(samplesIntegFltMoMa, features = 'Pdgfc', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
15 p13 <- FeaturePlot(samplesIntegFltMoMa, features = 'Ifit1', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
16 p14 <- FeaturePlot(samplesIntegFltMoMa, features = 'Mki67', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))

```

```

17 p15 <- FeaturePlot(samplesIntegFltMoMa, features = 'Itgax', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
18 p16 <- FeaturePlot(samplesIntegFltMoMa, features = 'Flt3', min.<-
  ↪ cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
19
20
21
22 plot_grid(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, ↪
  ↪ p14, p15, p16, ncol=4)

```

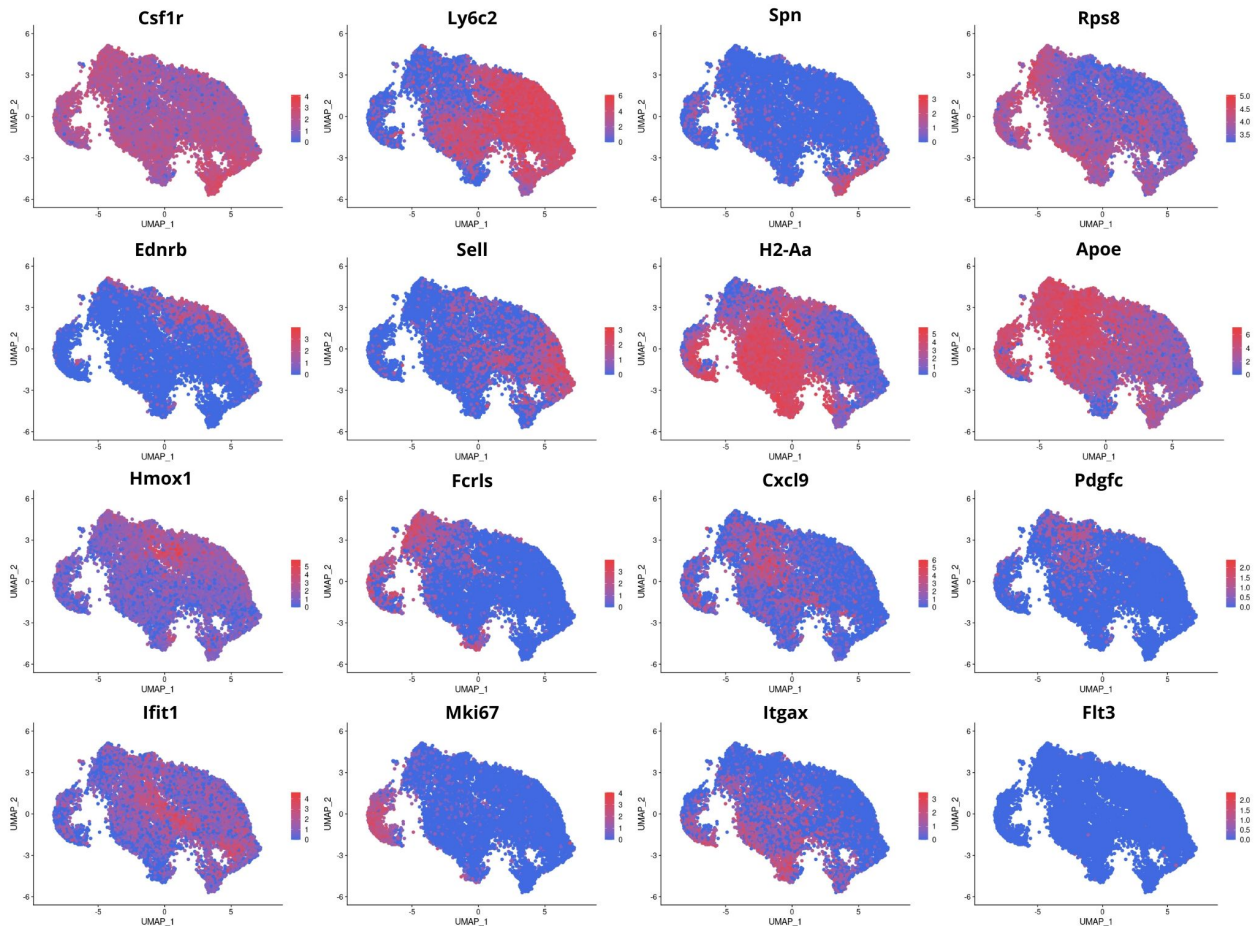


Figure D.20: Caption

Código D.11: R studio code extract – Monocyte/Macrophages trajectory analysis using Monocle 3

```

1 #--- MONOCLE 3 Trajectory ---#
2 #Transform Seurat object to Monocle object
3 monocle_object <- as.cell_data_set(samplesIntegFltMoMa)
4
5

```

```

6 #-- Retrieve Seurat information --#
7 #get cell metadata
8 colData(monocle_object)
9
10 #get gene metadata
11 fData(monocle_object)
12 rownames(fData(monocle_object))[1:10]
13 fData(monocle_object)$gente_short_name <- rownames(fData(monocle_↵
    ↵ object))
14 fData(monocle_object)
15
16 #get counts
17 counts(monocle_object)
18
19 #-- Retrive clusterization from Seurat ---#
20
21 #assign partitions
22 recreate.partition <- c(rep(1,length(monocle_↵
    ↵ object@colData@rownames)))
23 names(recreate.partition) <- monocle_object@colData@rownames
24 recreate.partition <- as.factor(recreate.partition)
25
26 monocle_object@clusters$UMAP$partitions <- recreate.partition
27
28 #assign cluster info from Seurat
29 list_cluster <- samplesIntegFltMoMa@active.ident
30 monocle_object@clusters$UMAP$clusters <- list_cluster
31
32 #assign UMAP coordinate - cell embeddings
33 monocle_object@int_colData@listData$reducedDims$UMAP <- ↵
    ↵ samplesIntegFltMoMa@reductions$umap@cell.embeddings
34
35 #-- Learn trajectory graph --#
36
37 monocle_object <- learn_graph(monocle_object, use_partition = TRUE↵
    ↵ )
38
39 #-- Order cells in pseudo-time --#
40
41 #assign begining of timeline
42 monocle_object <- order_cells(monocle_object, reduction_method = '↵
    ↵ UMAP', root_cells = colnames(monocle_object[,clusters(↵
    ↵ monocle_object)==4]))
43
44 #plot by pseudo-time

```



```

45 plot_cells(monocle_object, color_cells_by = 'pseudotime', label_↵
    ↵ groups_by_cluster = FALSE,
46     label_branch_points= FALSE, label_roots = FALSE, label_↵
    ↵ leaves = FALSE ,
47     cell_size = 1.5, trajectory_graph_color = "black", ↵
    ↵ trajectory_graph_segment_size = 1.5) + scale_color_gradient2(↵
    ↵ (high="brown2", low = "white")

```

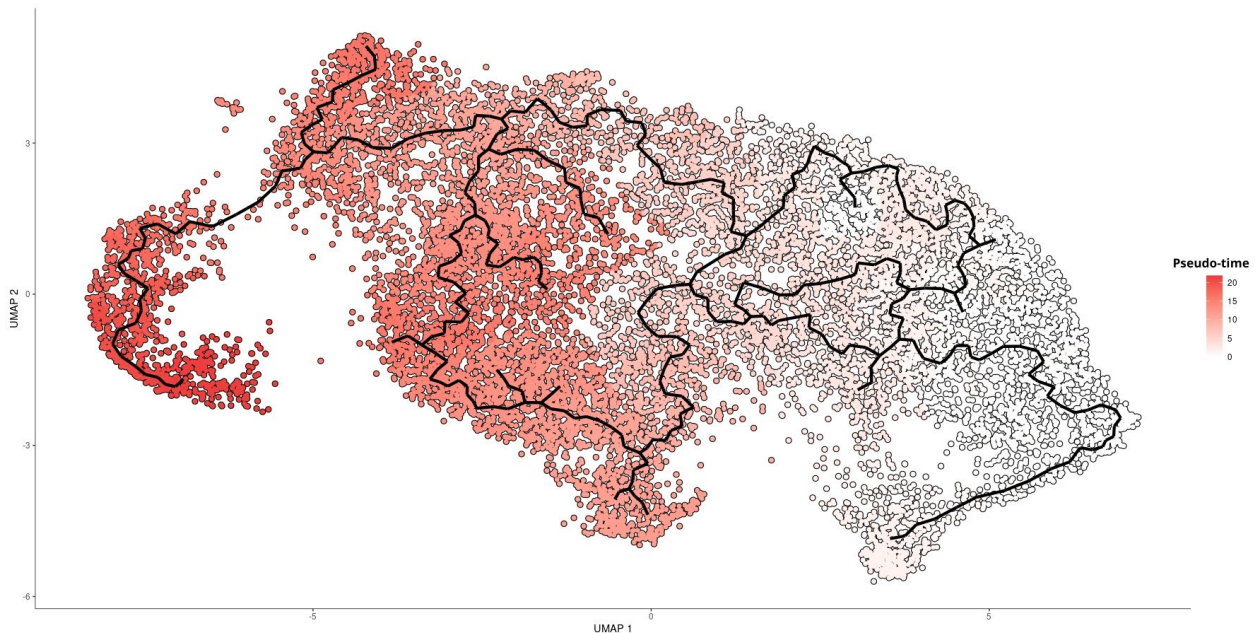


Figure D.21: Monocyte/Macrophages trajectory using Monocle3

Código D.12: R studio code extract – Monocyte/Macrophages subpopulation division

```

1 #-- New classification --#
2 samplesIntegFltMoMa <- RenameIdents(object = samplesIntegFltMoMa, ↵
    ↵ 0= Monocyte-Derived Mac,1=Rps8 Mono, 2=Ednrb Mono, 3=Fcrls ↵
    ↵ Mac, 4=Sell Mono,
3     5= Cycling Mac,6=Cxcl9 Mac, 7=↵
    ↵ Monocyte-Derived DC (Flt3 -), 8=Pdgfc Mac, 9=IFN Mac,
4     10=Mreg, 11=Resident Mono)
5
6 DimPlot(samplesIntegFltMoMa, reduction = "umap", label = FALSE, pt↵
    ↵ .size = 1.5, cols = my_colors, label.size = 6) + ggtitle('↵
    ↵ Monocyte/Macrophages subpopulations')

```

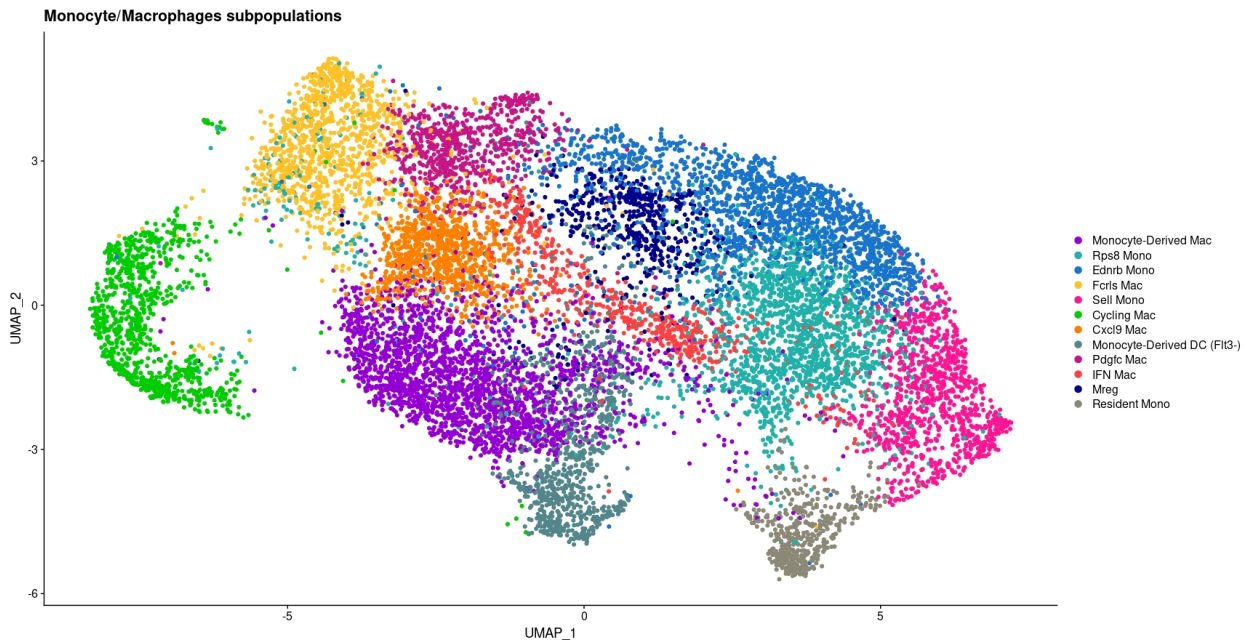



Figure D.22: Monocyte/Macrophages subpopulations

Código D.13: R studio code extract – Monocyte/Macrophages Barplots

```

1
2 samplesIntegFltMoMa[["Subpopulation"]] <- samplesIntegFltMoMa@meta↵
↵ .data$integrated_snn_res.0.7
3 dittoBarPlot(object = samplesIntegFltMoMa, scale = "percent", var ↵
↵ = "Subpopulation", group.by = "Treatment", color.panel = my_↵
↵ colors, x.labels = c("Untreated", "Treated"), xlab = "↵
↵ Condition", main = "")

```

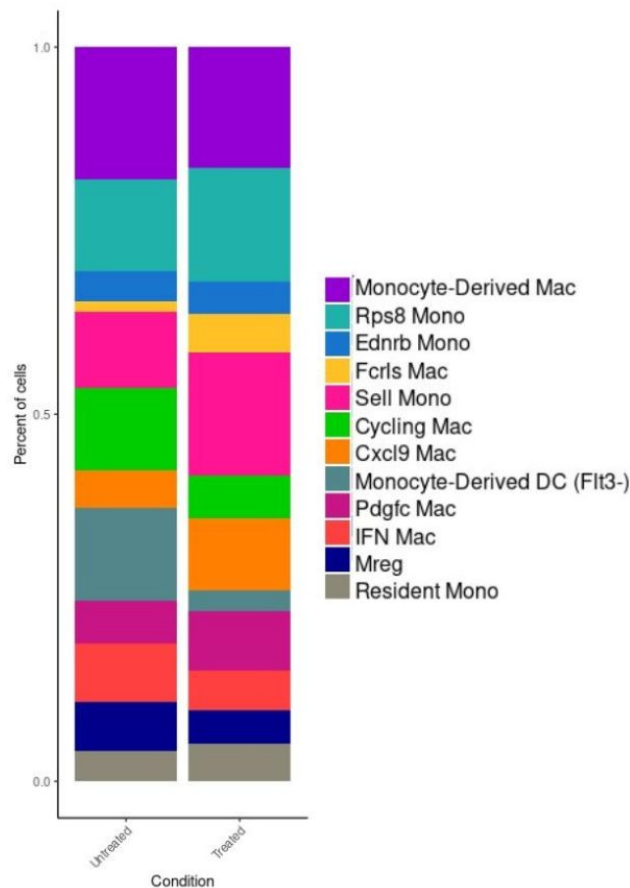


Figure D.23: Barplot of Monocyte/Macrophages subpopulations percentage by Conditions

Código D.14: R studio code extract – Macrophage subset separation

```

1
2 #- Macrophage subset -#
3 samplesIntegFltMa <- subset(samplesIntegFltMoMa, idents=c("↔
↔ Monocyte-Derived Mac", "Fcrls Mac", "Cycling Mac", "Cxcl9 Mac↔
↔ ", "Pdgfc Mac", "IFN Mac", "Mreg"), invert=FALSE)
4
5 my_colorsMa=c("darkviolet", "goldenrod1", "green3", "darkorange1", ↔
↔ "mediumvioletred", "brown1", "blue4")
6
7 DimPlot(samplesIntegFltMa, reduction = "umap", label = FALSE, pt.↔
↔ size = 1.5, cols = my_colorsMa, label.size = 6) + ggtitle('↔
↔ Macrophages subpopulations')
8
9 #- Divided by Condition -#
10 samplesIntegFltMa.list <- SplitObject(samplesIntegFltMa, split.by ↔
↔ = "Treatment")

```

```

11
12 Ma_Untreated <- samplesIntegFltMa.list[[1]]
13 Ma_Treated <- samplesIntegFltMa.list[[2]]
14
15 saveRDS(Ma_Untreated, "Ma_Untreated.rds")
16 saveRDS(Ma_Treated, "Ma_Treated.rds")

```

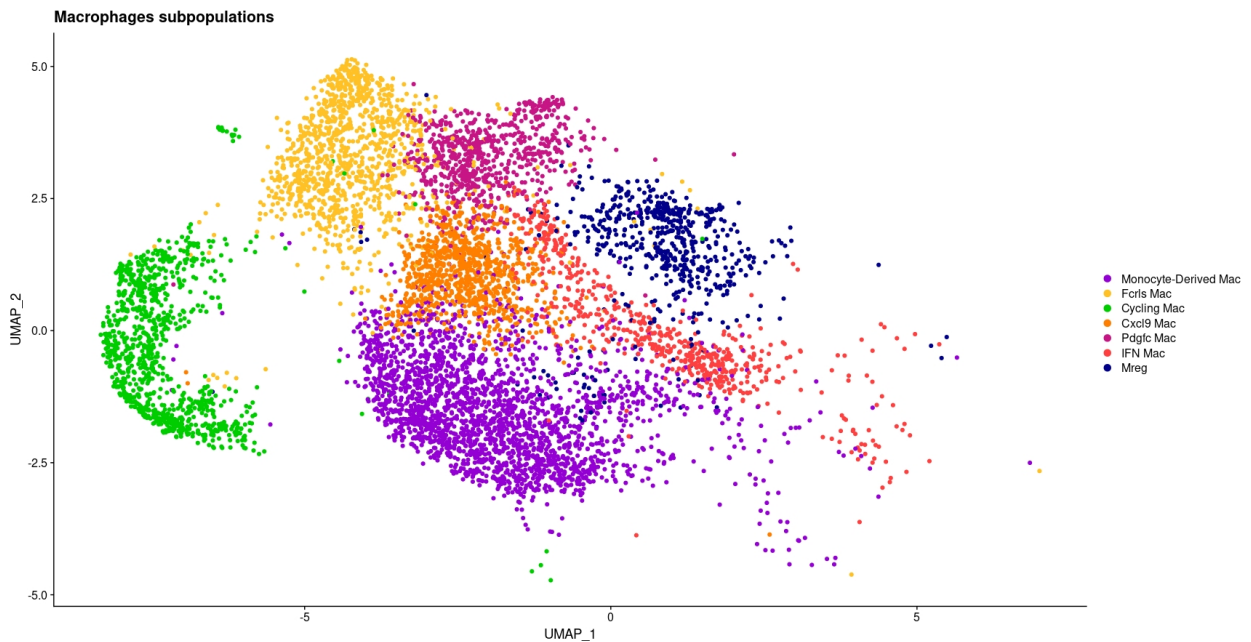


Figure D.24: Macrophages subpopulations

Código D.15: R studio code extract – Macrophages Spp1 expression analysis

```

1
2 DefaultAssay(object = samplesIntegFltMa) <- "SCT"
3
4 #- All macrophages -#
5 FeaturePlot(samplesIntegFltMa, features = 'Spp1', min.cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2"))
6
7 #- Untreated -#
8 FeaturePlot(Ma_Untreated, features = 'Spp1', min.cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2")) + ggtitle("Untreated Spp1")
9
10 #- Treated -#
11 FeaturePlot(Ma_Treated, features = 'Spp1', min.cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2")) + ggtitle("Treated Spp1")

```

```

12     ↪ Spp1")
13
14 #- Comparison -#
15 VlnPlot(samplesIntegFltMa, features = "Spp1", group.by = "↪
16     ↪ Treatment", pt.size = 0) +
17     scale_x_discrete(labels = c("Untreated" = "Untreated",
18                               "Treated" = "Treated"))

```

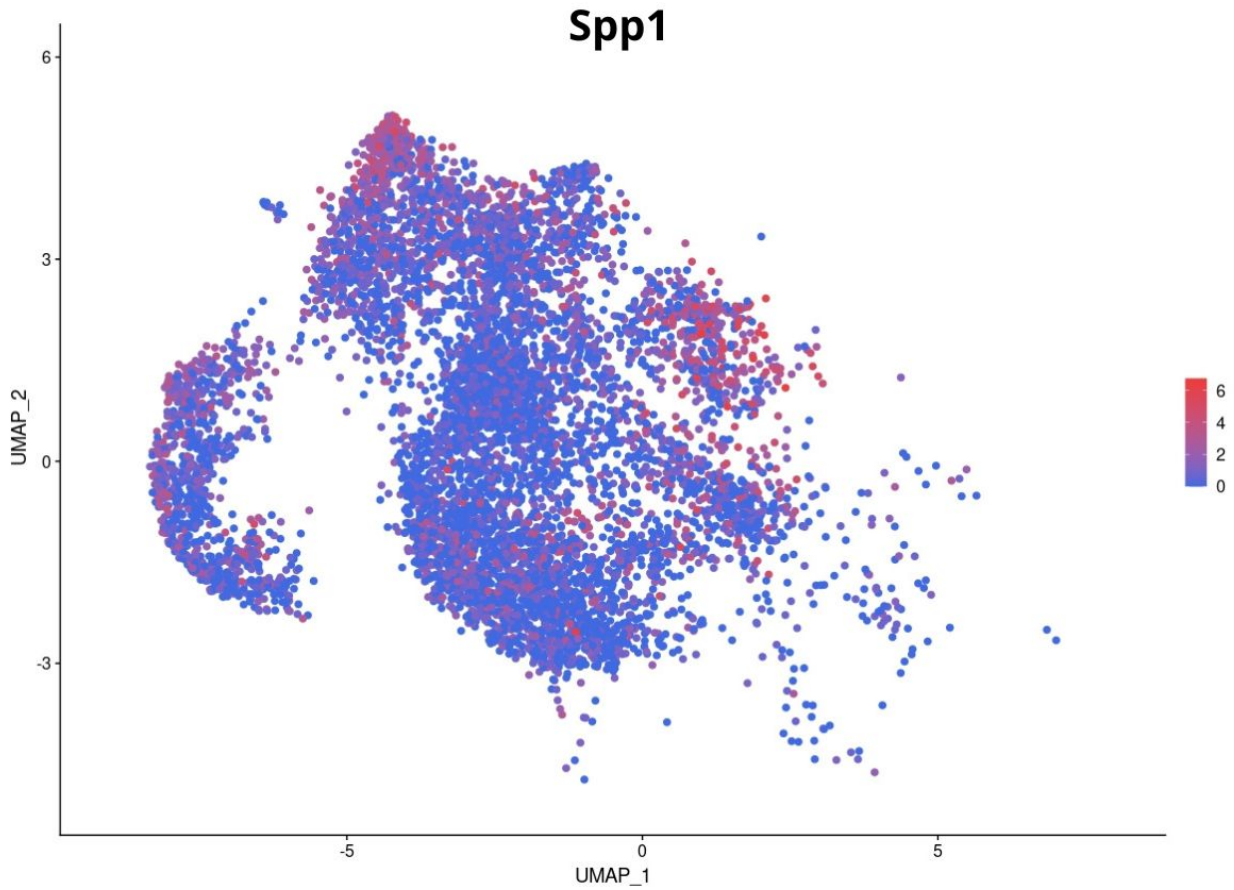


Figure D.25: Spp1 expression level in Macrophages

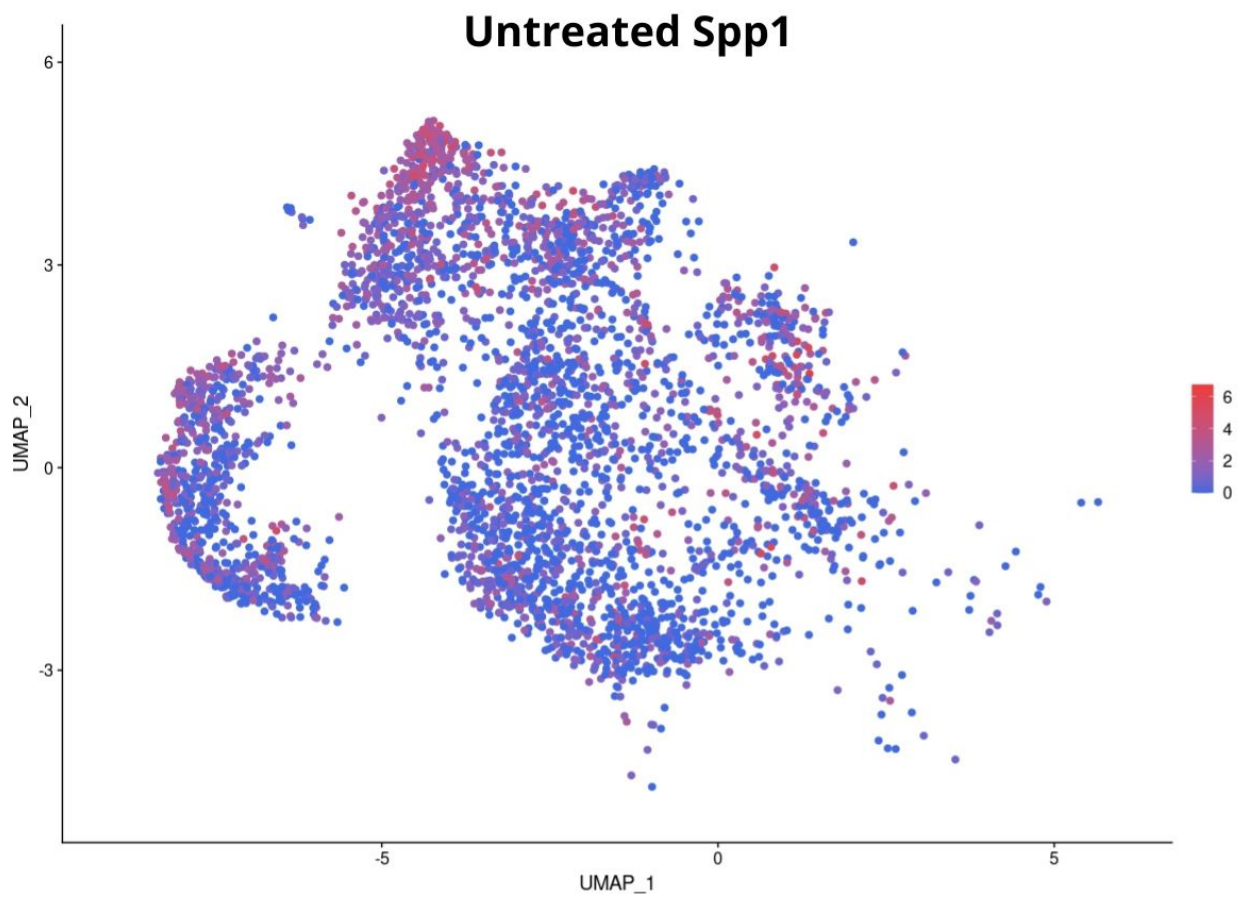


Figure D.26: Spp1 expression level in untreated Macrophages

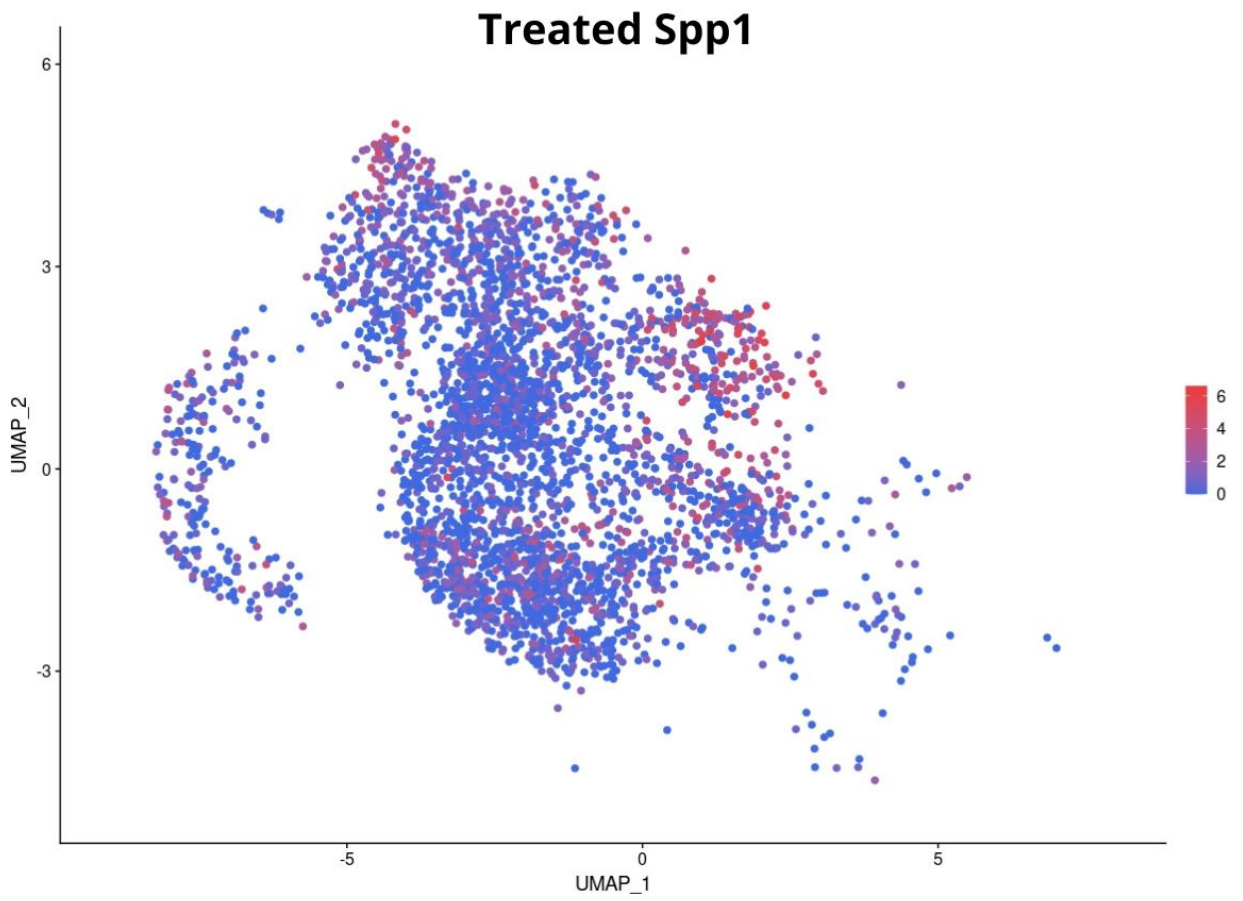


Figure D.27: Spp1 expression level in treated Macrophages

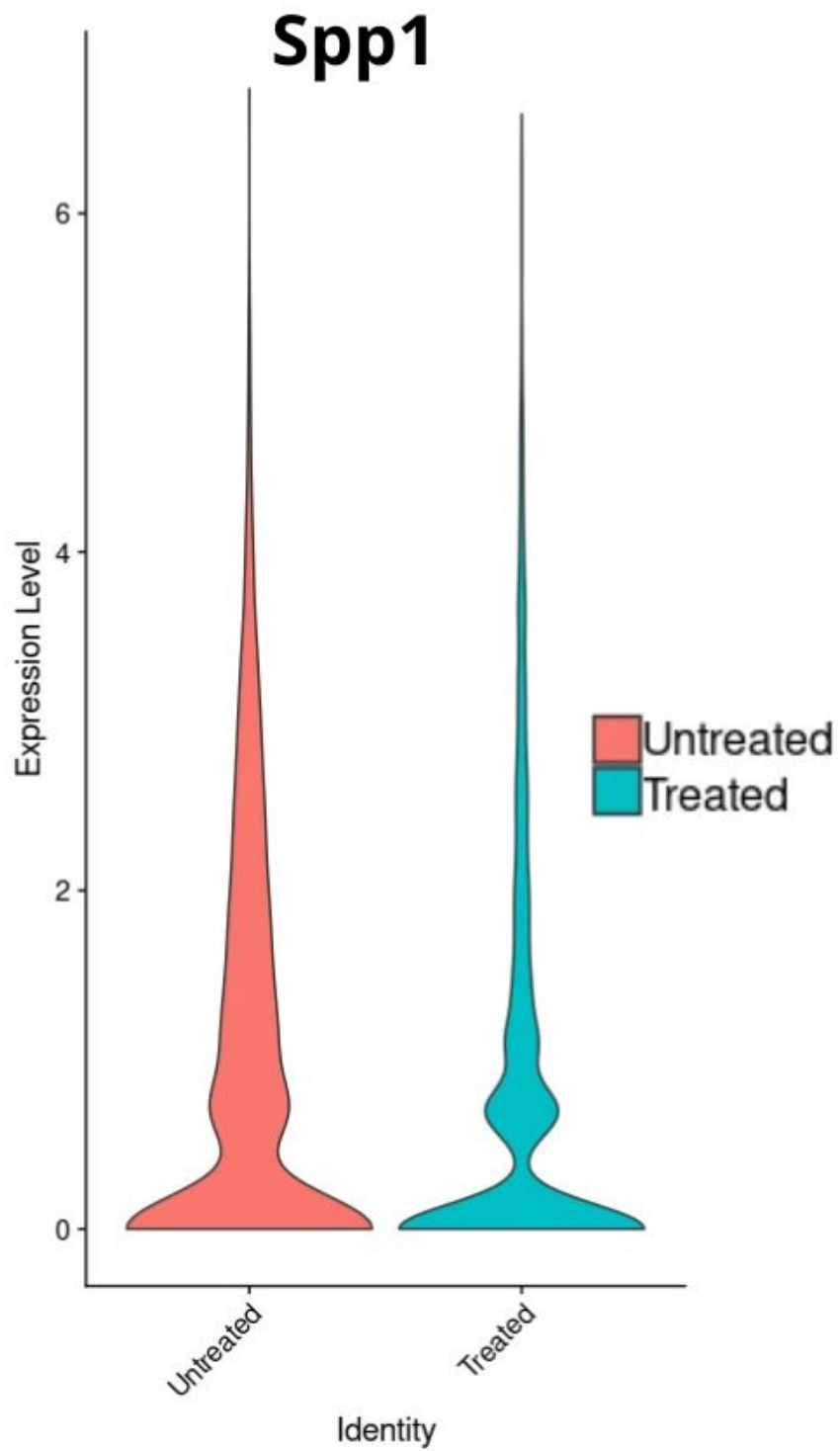


Figure D.28: Macrophages Spp1 violin plot by Condition

Código D.16: R studio code extract – Macrophages Cxcl9 expression analysis

```

2 DefaultAssay(object = samplesIntegFltMa) <- "SCT"
3
4 #- All macrophages -#
5 FeaturePlot(samplesIntegFltMa, features = 'Cxcl9', min.cutoff = 0, ←
  ↪ pt.size = 2, cols = c("royalblue", "brown2"))
6
7 #- Untreated -#
8 FeaturePlot(Ma_Untreated, features = 'Cxcl9', min.cutoff = 0, pt. ←
  ↪ size = 2, cols = c("royalblue", "brown2")) + ggtitle(" ←
  ↪ Untreated Cxcl9")
9
10 #- Treated -#
11 FeaturePlot(Ma_Treated, features = 'Cxcl9', min.cutoff = 0, pt. ←
  ↪ size = 2, cols = c("royalblue", "brown2")) + ggtitle(" ←
  ↪ Treated Cxcl9")
12
13
14 #- Comparison -#
15 VlnPlot(samplesIntegFltMa, features = "Cxcl9", group.by = " ←
  ↪ Treatment", pt.size = 0) +
16   scale_x_discrete(labels = c("Untreated" = "Untreated",
17     ↪ "Treated" = "Treated"))

```

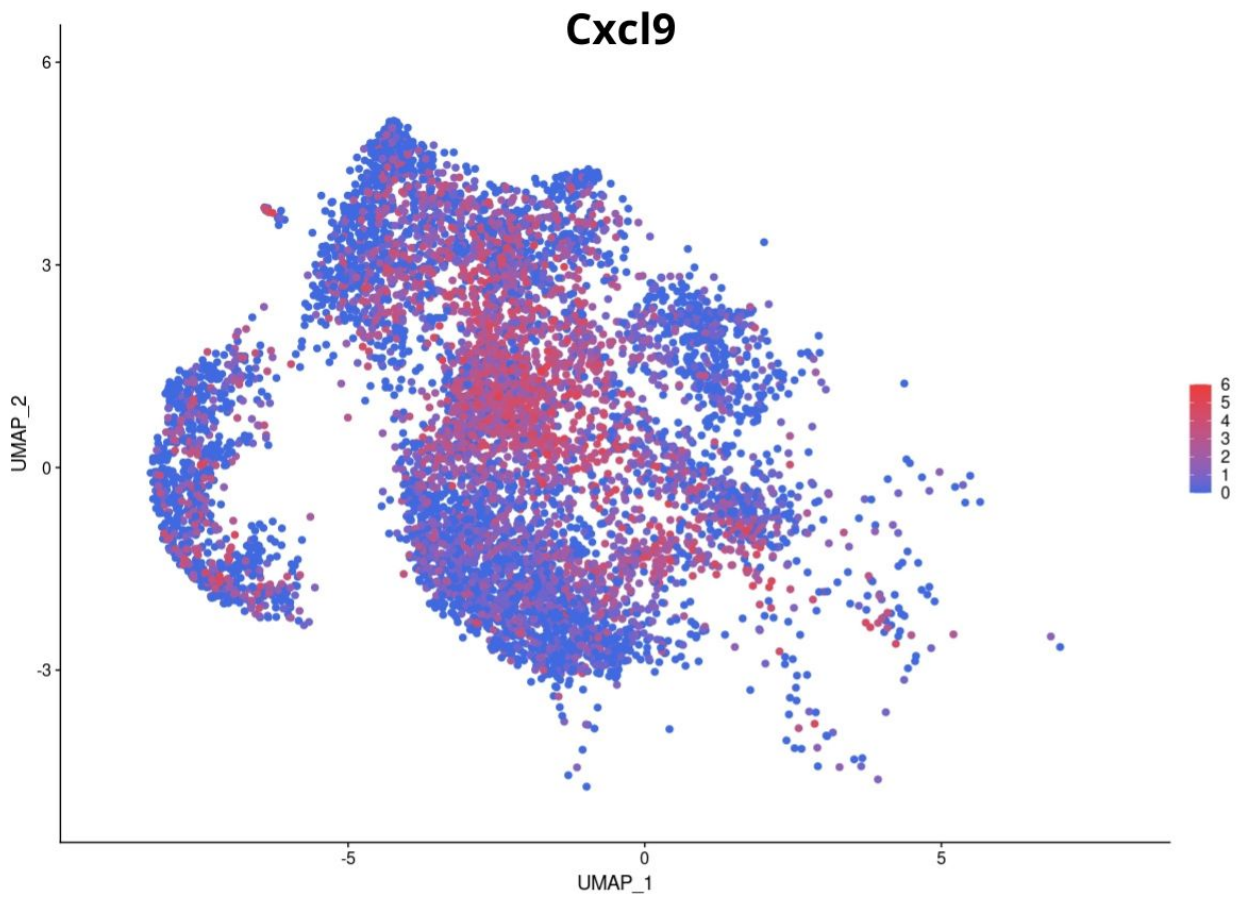



Figure D.29: Cxcl9 expression level in Macrophages

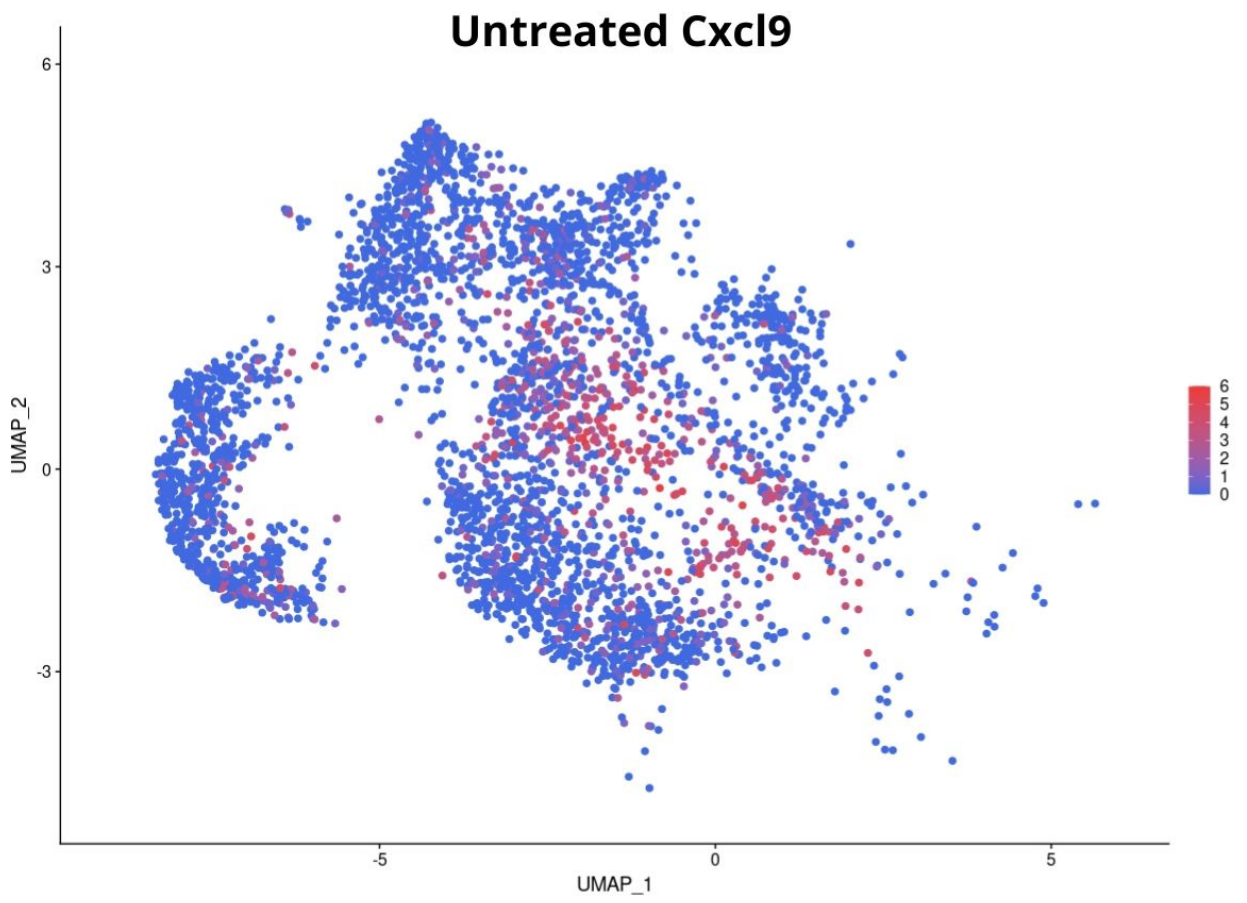


Figure D.30: Cxcl9 expression level in untreated Macrophages

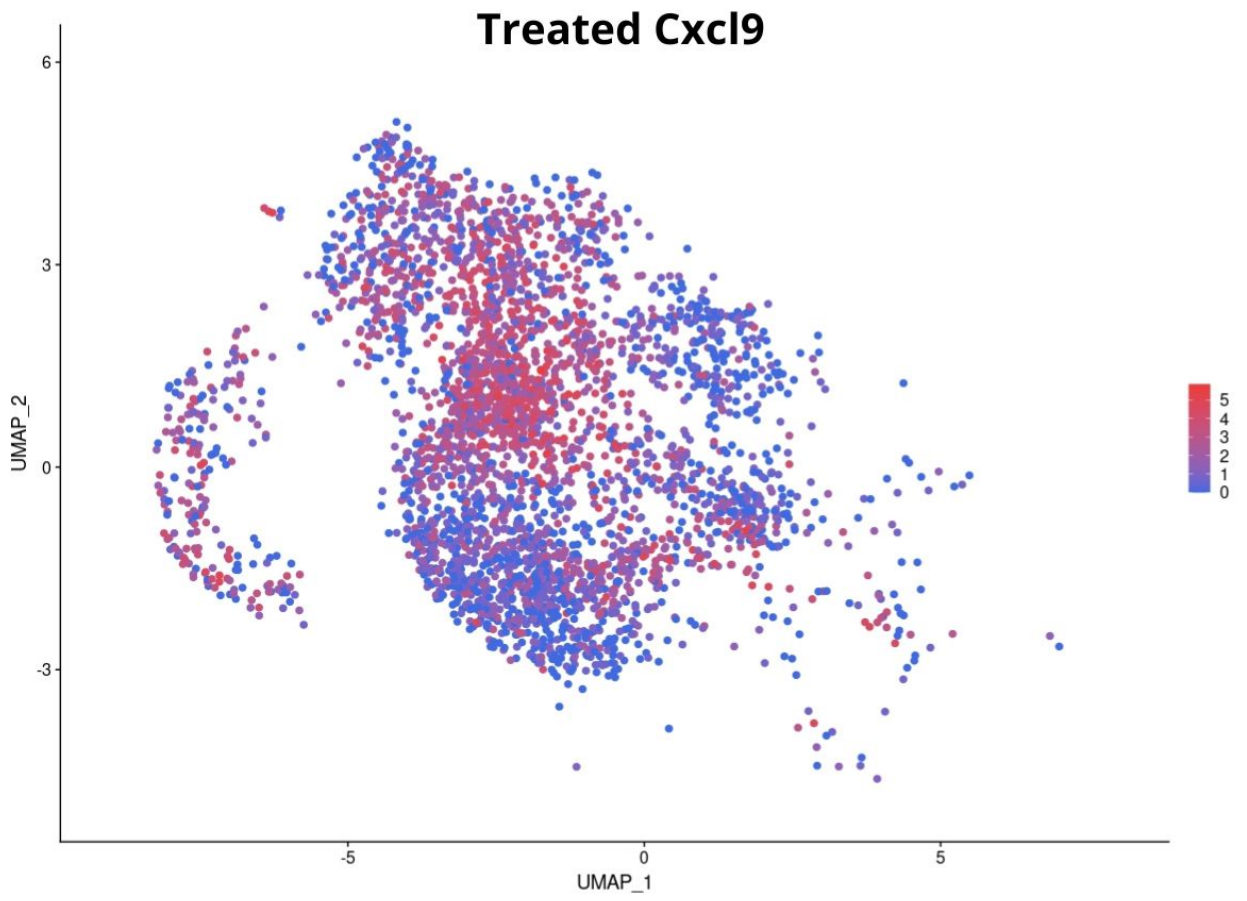


Figure D.31: Cxcl9 expression level in treated Macrophages

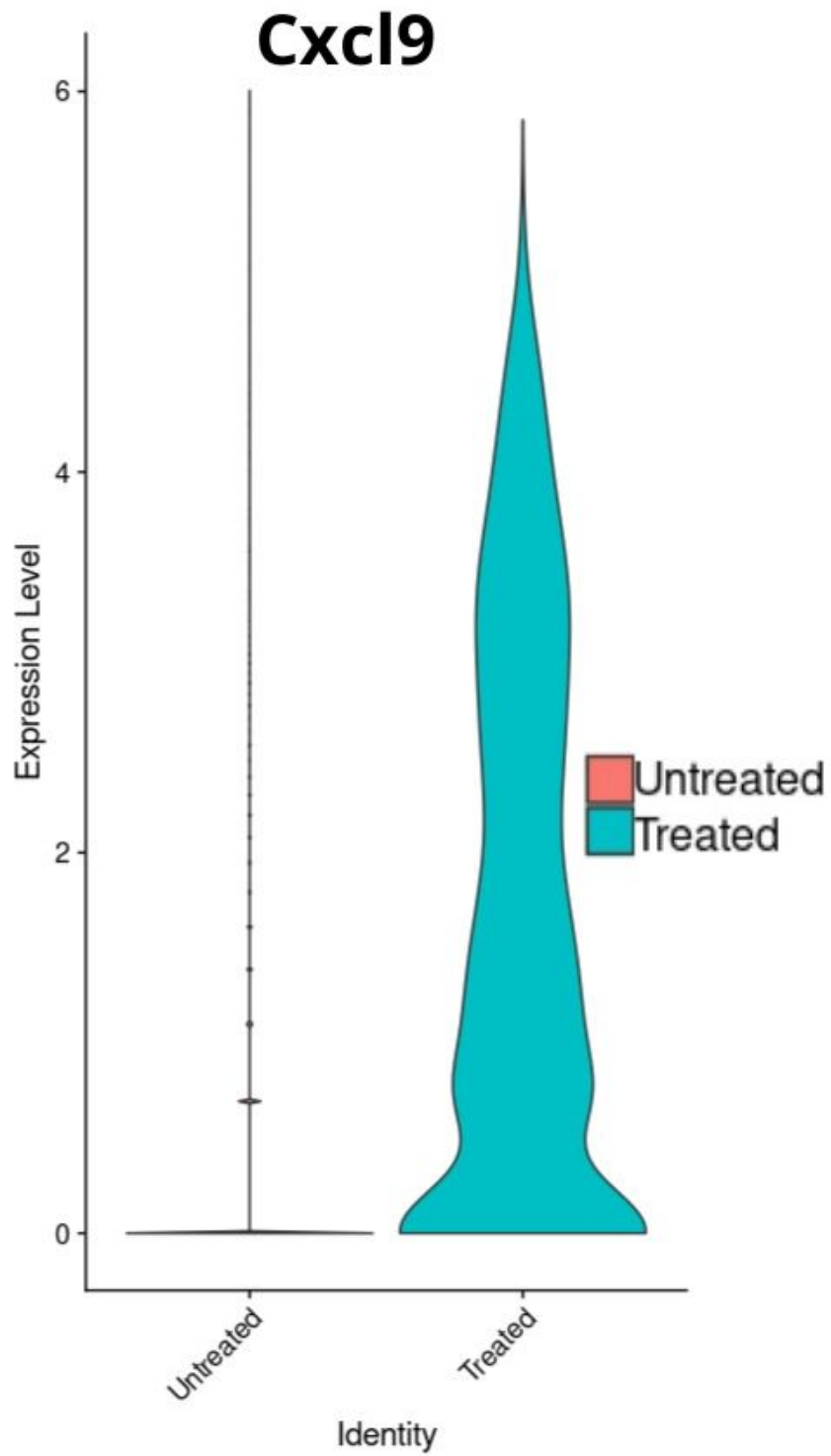


Figure D.32: Macrophages Cxcl9 violin plot by Condition

Código D.17: R studio code extract – Macrophages Cxcl10 expression analysis

```

2 DefaultAssay(object = samplesIntegFltMa) <- "SCT"
3
4 #- All macrophages -#
5 FeaturePlot(samplesIntegFltMa, features = 'Cxcl10', min.cutoff = <-
  ↪ 0, pt.size = 2, cols = c("royalblue", "brown2"))
6
7 #- Untreated -#
8 FeaturePlot(Ma_Untreated, features = 'Cxcl10', min.cutoff = 0, pt.<-
  ↪ size = 2, cols = c("royalblue", "brown2")) + ggtitle("<-
  ↪ Untreated Cxcl10")
9
10 #- Treated -#
11 FeaturePlot(Ma_Treated, features = 'Cxcl10', min.cutoff = 0, pt.<-
  ↪ size = 2, cols = c("royalblue", "brown2")) + ggtitle("<-
  ↪ Treated Cxcl10")
12
13
14 #- Comparison -#
15 VlnPlot(samplesIntegFltMa, features = "Cxcl10", group.by = "<-
  ↪ Treatment", pt.size = 0) +
16   scale_x_discrete(labels = c("Untreated" = "Untreated",
17     ↪ "Treated" = "Treated"))

```

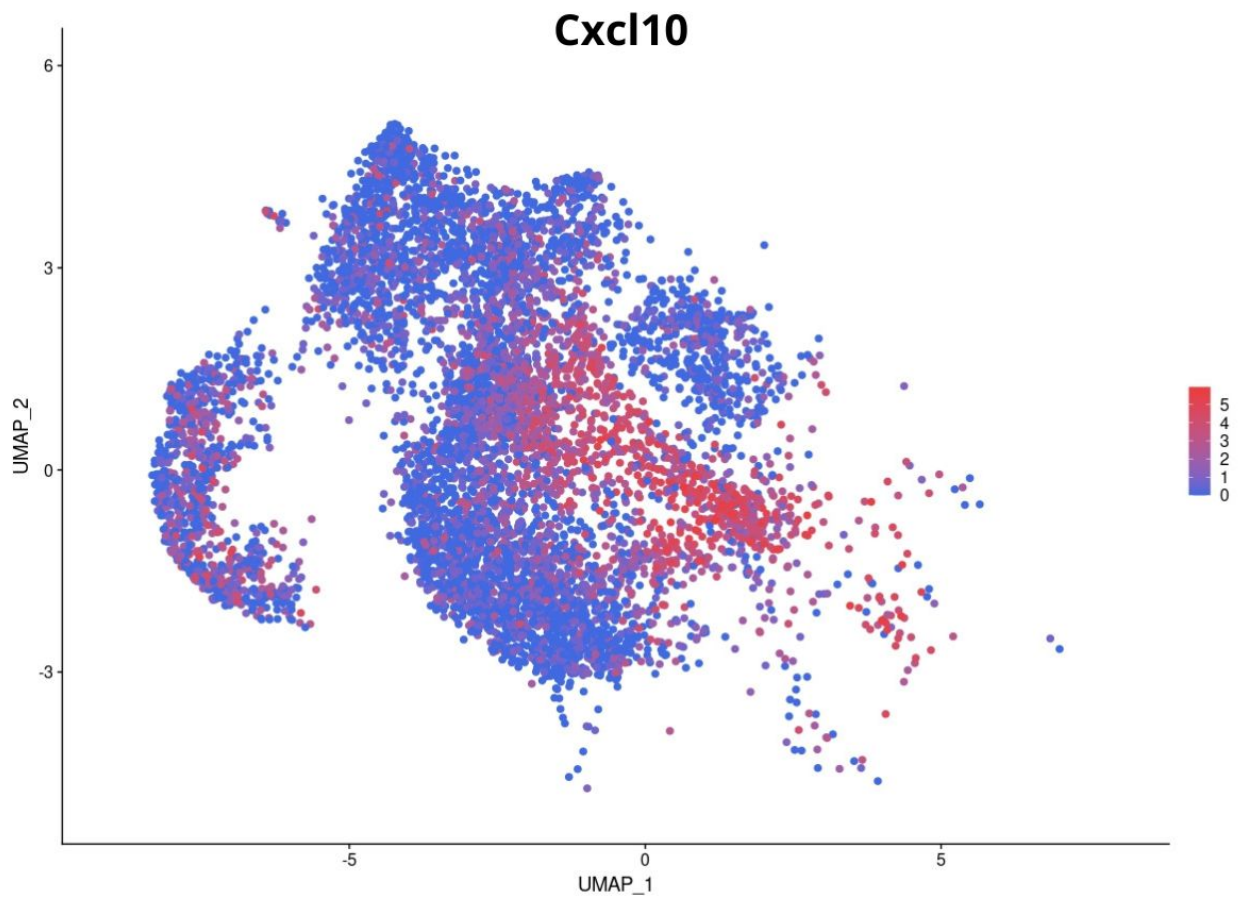


Figure D.33: Cxcl10 expression level in Macrophages

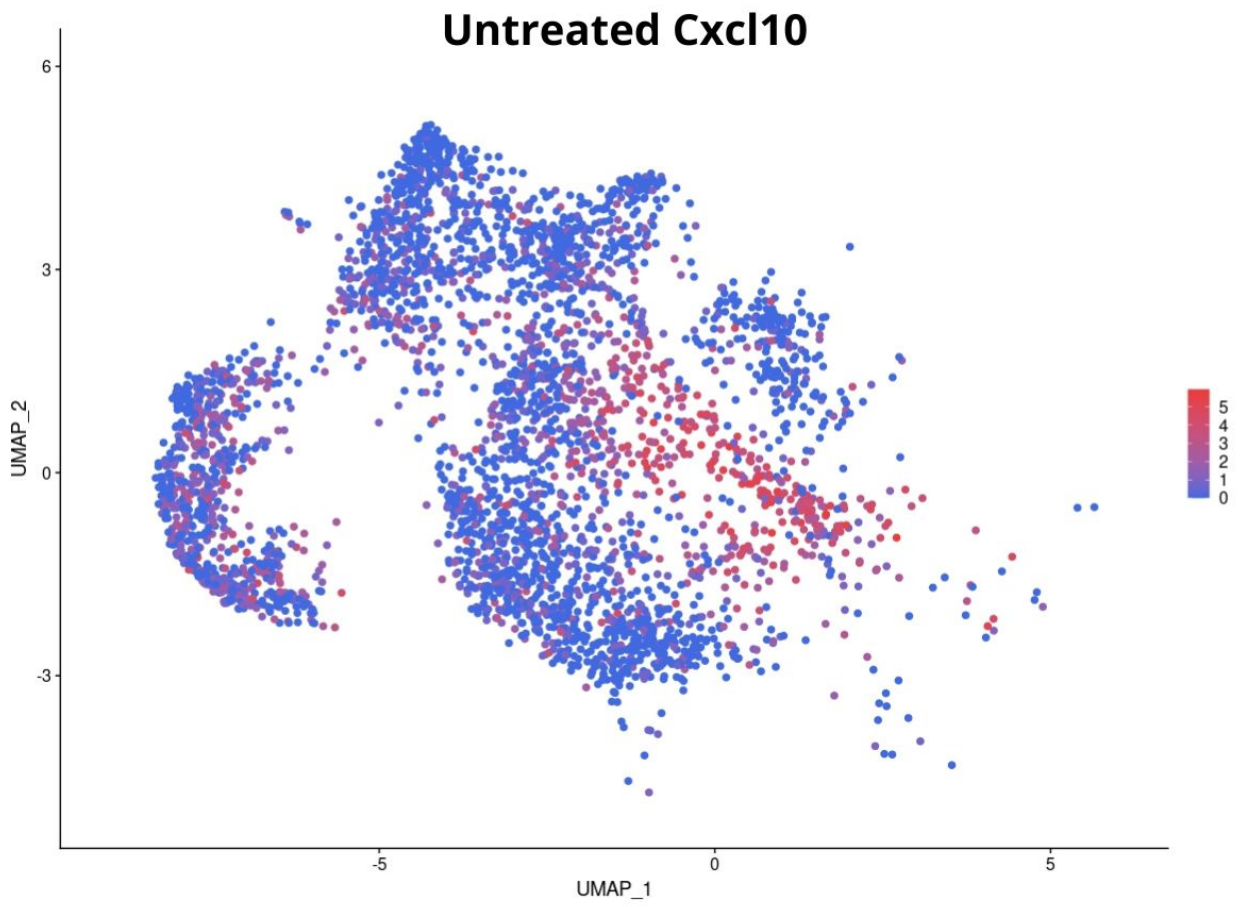


Figure D.34: Cxcl10 expression level in untreated Macrophages

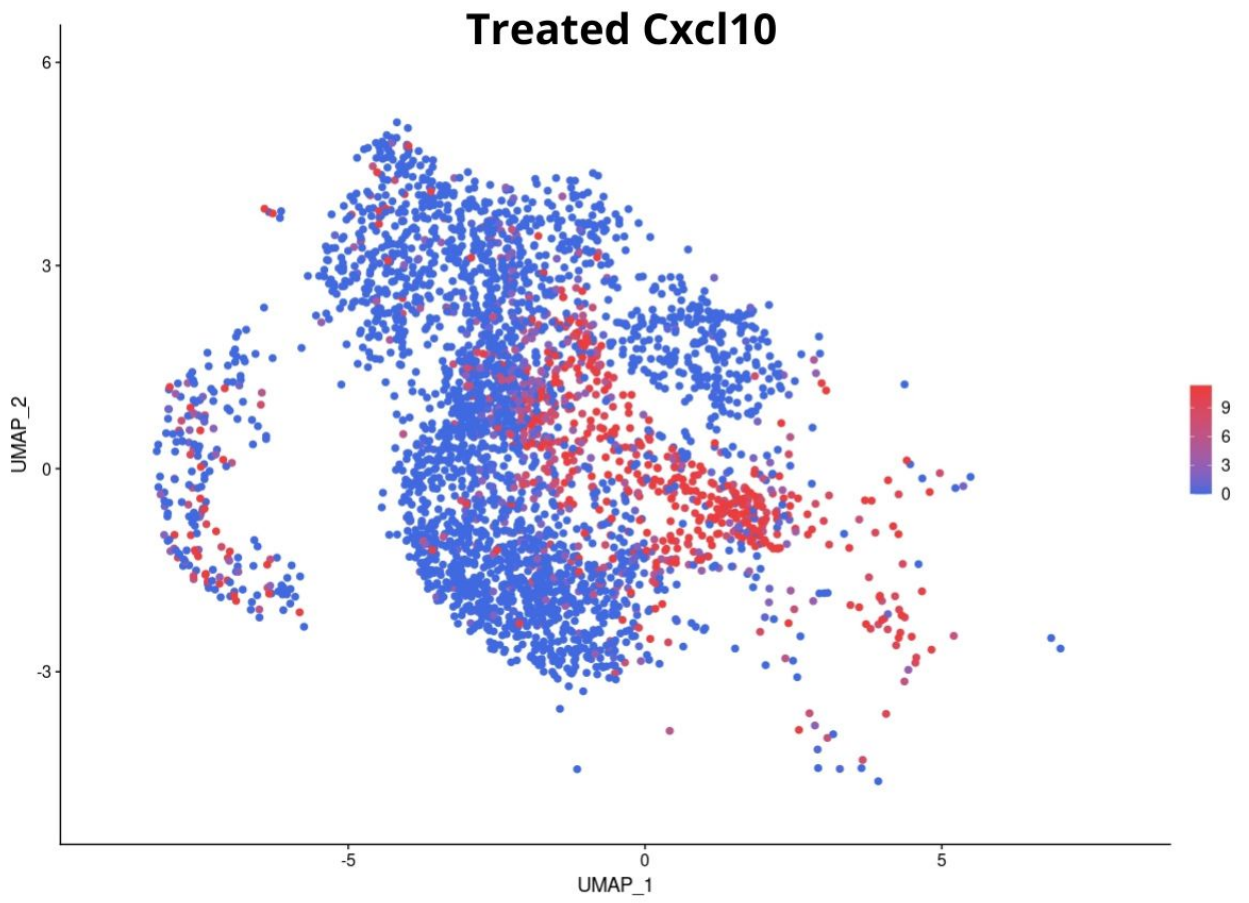


Figure D.35: Cxcl10 expression level in treated Macrophages

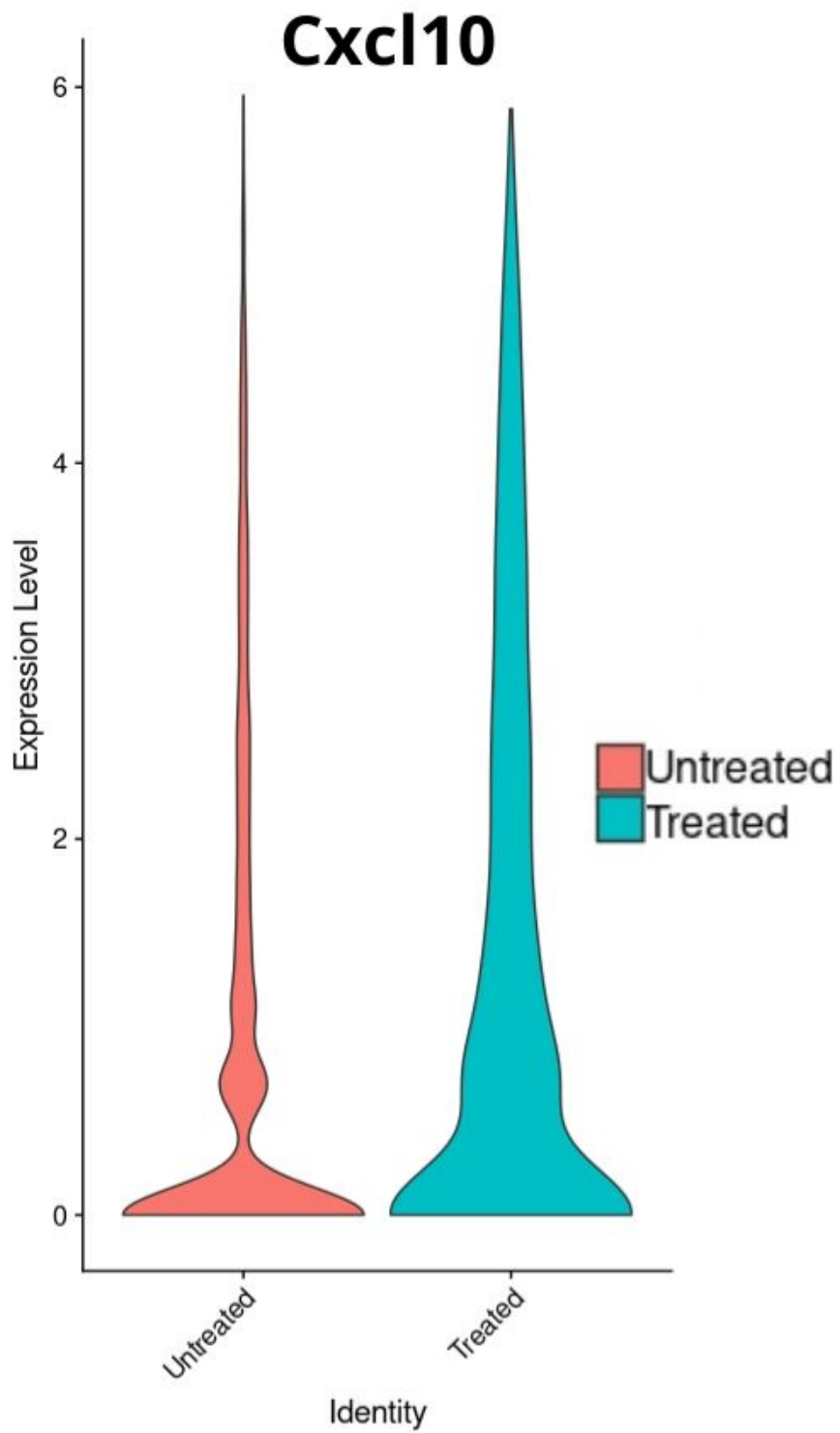


Figure D.36: Macrophages Cxcl10 violin plot by Condition

Annex E. Granulocyte cells mice analysis code

Código E.1: R studio code extract – Merge of both granulocyte objects

```
1 #----- Granulocytes analysis -----#
2
3 #-- Merge --#
4 #Merging previous granulocyte file with the extracted granulocytes↵
  ↵ of the MoMa dataset
5 samplesMrg <- merge(samplesIntegFltGranulocyteslocytes , y = ↵
  ↵ ContaminantGranu , add.cell.ids=c("↵
  ↵ samplesIntegFltGranulocyteslocytes" , y = "ContaminantGranu")↵
  ↵ , project="scRNA-seq")
6
7 samplesMrg <- SCTransform(object = samplesMrg, vars.to.regress = "↵
  ↵ percent.mt" , verbose = FALSE, variable.features.n = 5000)
```

Código E.2: R studio code extract – Granulocytes Integration

```
1 #--- Integration ---#
2
3 samplesGranu.list <- list(Untreated1, Untreated2, Treated1, ↵
  ↵ Treated2)
4
5 #- Normalice samples -#
6 samplesGranu.list <- lapply(X= samplesGranu.list , FUN = function(x↵
  ↵ ){
7   x <- SCTransform(object = x, vars.to.regress = "percent.mt" , ↵
  ↵ verbose = FALSE, variable.features.n = 5000)})
8
9 #- Select integration features and prepare for integration -#
10 samplesGranu.features <- SelectIntegrationFeatures(object.list = ↵
  ↵ samplesGranu.list )
11 samplesGranu.list <- PrepSCTIntegration(object.list = samplesGranu↵
  ↵ .list , anchor.features = samplesGranu.features)
12
13 #- Select integration anchors -#
14 samplesGranu.anchors <- FindIntegrationAnchors(object.list = ↵
  ↵ samplesGranu.list , normalization.method = "SCT" , anchor.↵
  ↵ features = samplesGranu.features)
15 save(samplesGranu.anchors , file=file.path("~/", "samplesGranu.↵
  ↵ anchors.rdata"))
16
17 #- Integrate -#
```

```

18 samplesIntegFltGranulocytes <- IntegrateData(anchorset = ↵
    ↵ samplesGranu.anchors, normalization.method = "SCT")
19 saveRDS(samplesIntegFltGranulocytes, "samplesIntegFltGranulocytes.↵
    ↵ rds")

```

Código E.3: R studio code extract – Granulocytes linear reduction

```

1 #- PCA reduction -#
2 samplesIntegFltGranulocytes <- RunPCA(samplesIntegFltGranulocytes, ↵
    ↵ npcs = 50, verbose=F)
3
4 #Elbow plot
5 ElbowPlot(samplesIntegFltGranulocytes, ndims = 50, reduction = "↵
    ↵ pca") + ggtitle("Elbow plot Integrated Granu data")

```

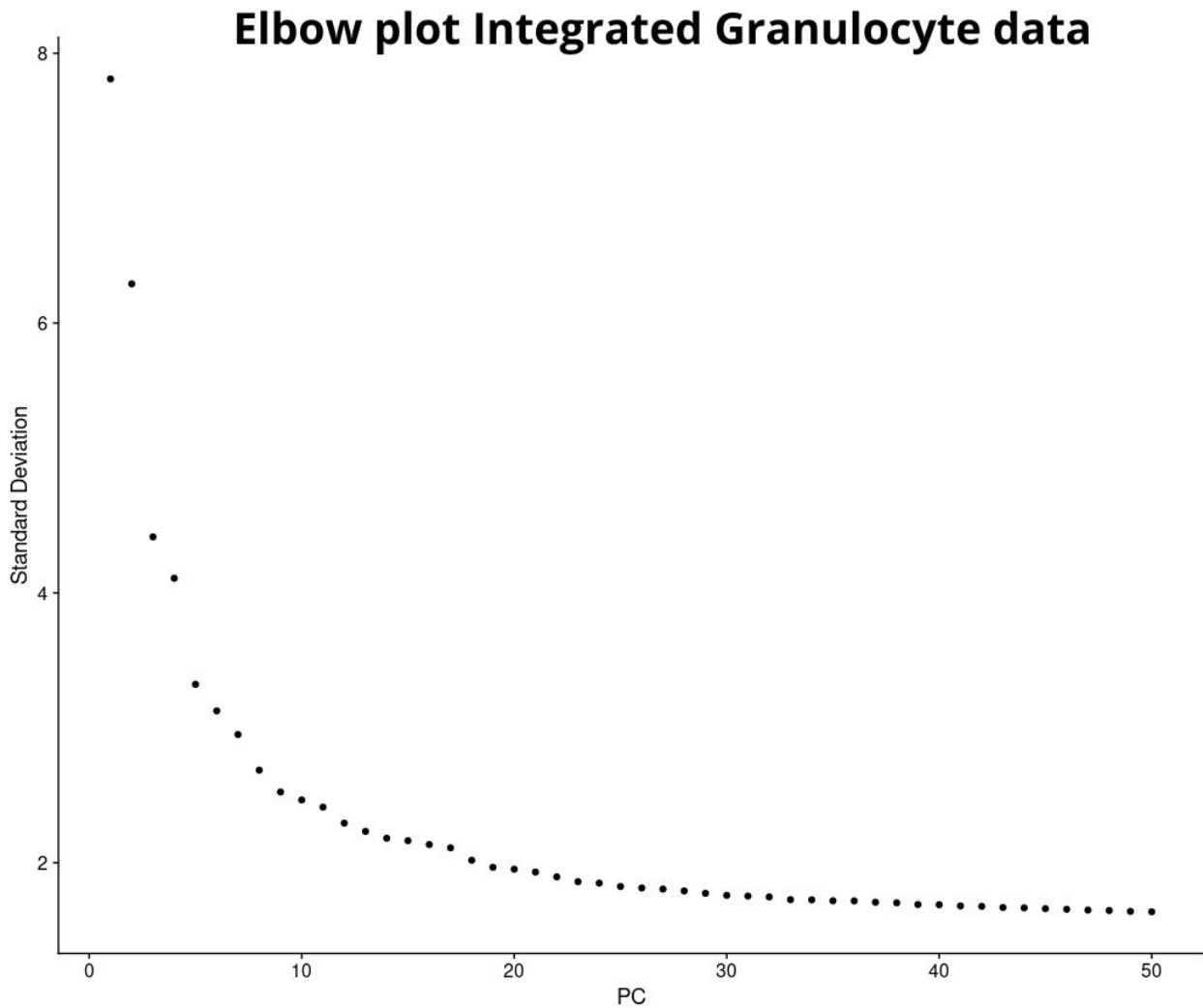


Figure E.1: Elbow plot Granulocytes

Código E.4: R studio code extract – Granulocytes non-linear reduction

```
1 #- UMAP Reduction -#
2 samplesIntegFltGranulocytes <- RunUMAP(samplesIntegFltGranulocytes , <-
  ↪ dims=1:40, verbose=F)
3
4 #UMAP by Condition
5 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", group.by <-
  ↪ = 'Treatment', pt.size = 1.5) + ggtitle("Integrated <-
  ↪ Granulocyte UMAP divided by Condition") + scale_color_<-
  ↪ manual(labels = c("Untreated", "Treated"), values = c("#<-
  ↪ F8766D", "#00bfc4"))
6
7 #UMAP by Sample
8 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", group.by <-
  ↪ = 'orig.ident', pt.size = 1.5) + ggtitle("Integrated <-
  ↪ Granulocyte UMAP divided by Sample") + scale_color_manual(<-
  ↪ labels = c("Untreated1", "Untreated2", "Treated1", "Treated2" <-
  ↪ ), values = my_colors)
```

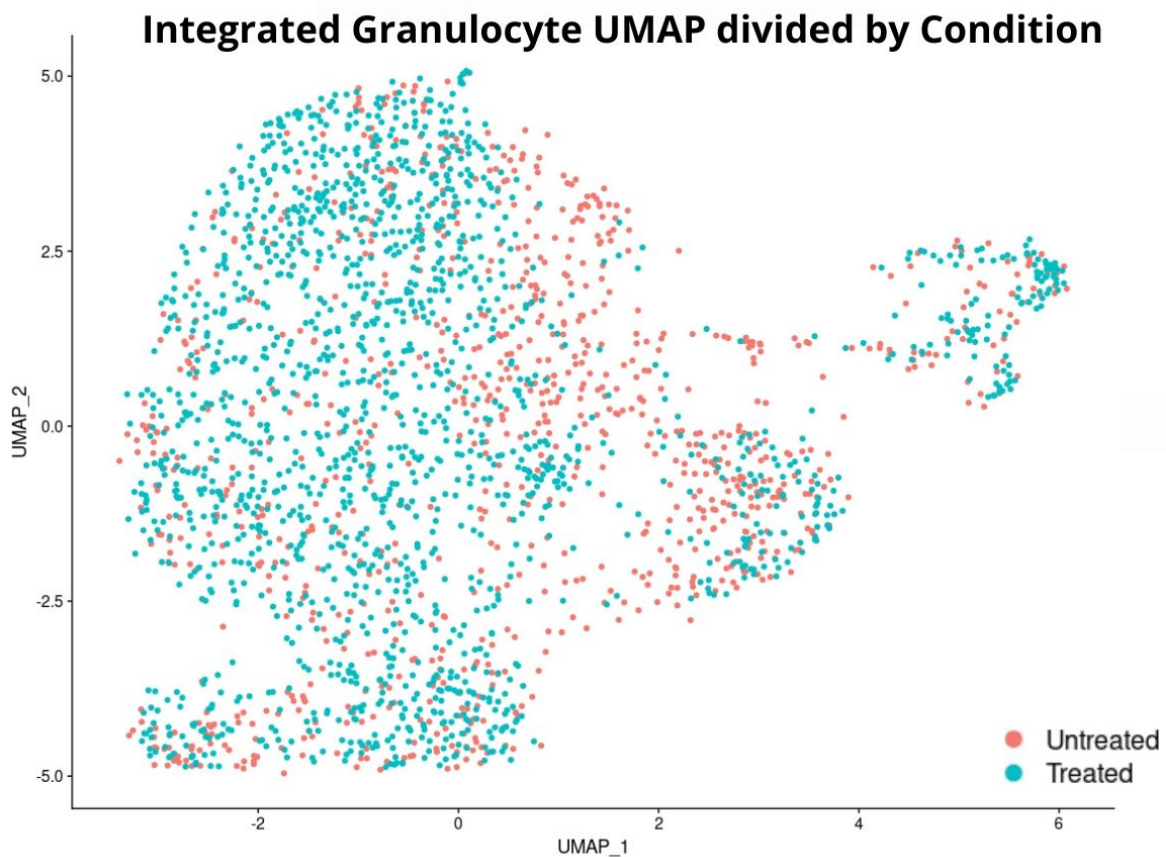


Figure E.2: UMAP of Granulocytes divided by Condition

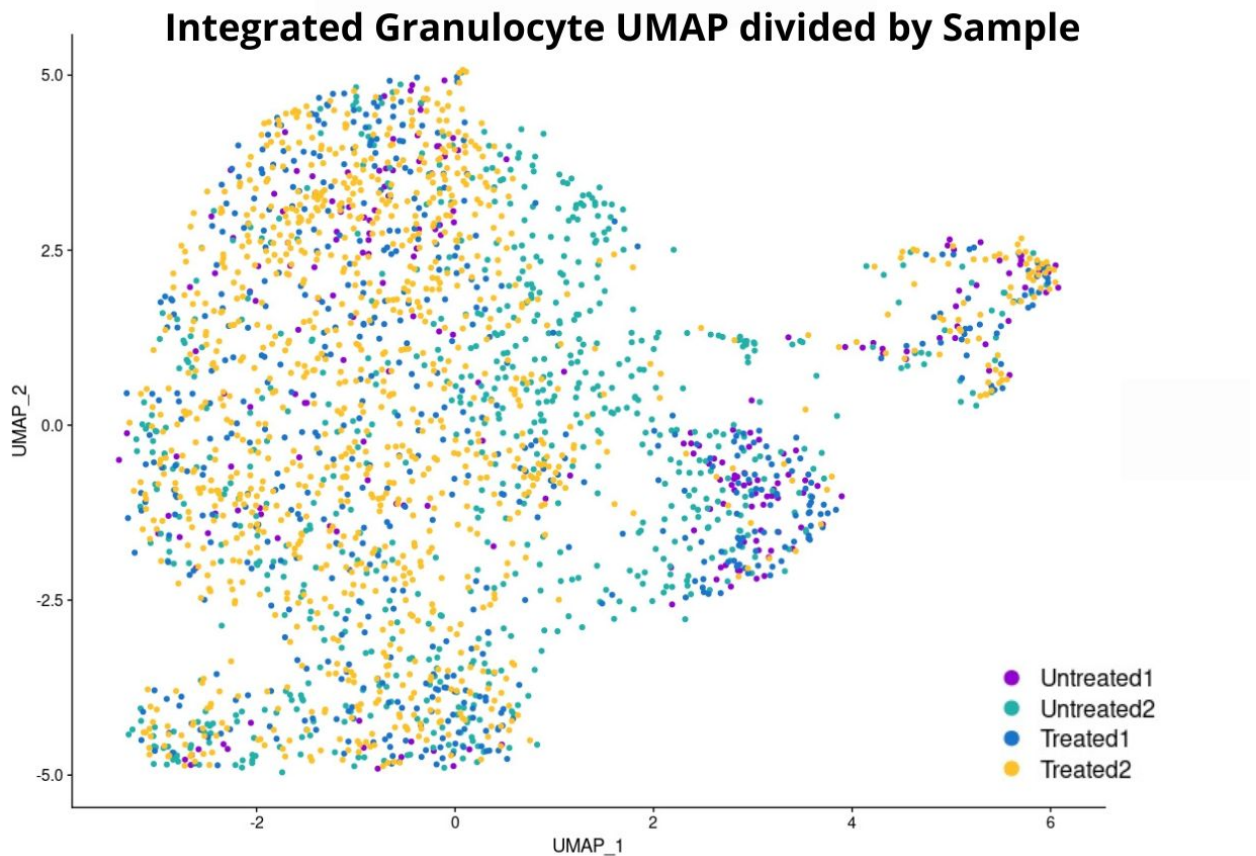


Figure E.3: UMAP of Granulocytes divided by Sample

Código E.5: R studio code extract – Granulocytes clustering

```

1 #--- Clustering ---#
2 DefaultAssay(object = samplesIntegFltGranulocytes) <- "integrated"
3
4 samplesIntegFltGranulocytes <- FindNeighbors(↵
  ↵ samplesIntegFltGranulocytes, dims = 1:50, verbose = FALSE)
5
6 samplesIntegFltGranulocytes <- FinGranulusters(↵
  ↵ samplesIntegFltGranulocytes, resolution = 0)
7 DimPlot(samplesIntegFltGranulocytes, reduction = "umap", label = ↵
  ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution 0↵
  ↵ Granulocytes')
8
9 samplesIntegFltGranulocytes <- FinGranulusters(↵
  ↵ samplesIntegFltGranulocytes, resolution = 0.1)
10 DimPlot(samplesIntegFltGranulocytes, reduction = "umap", label = ↵
  ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
  ↵ 0.1 Granulocytes')
11

```

```

12 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.2)
13 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.2 Granulocytes')
14
15 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.3)
16 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.3 Granulocytes')
17
18 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.4)
19 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.4 Granulocytes')
20
21 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.5)
22 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.5 Granulocytes')
23
24 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.6)
25 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.6 Granulocytes')
26
27 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.7)
28 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.7 Granulocytes')
29
30 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.8)
31 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.8 Granulocytes')
32
33 samplesIntegFltGranulocytes <- FinGranulusters(↵
    ↵ samplesIntegFltGranulocytes , resolution = 0.9)
34 DimPlot(samplesIntegFltGranulocytes , reduction = "umap", label = ↵

```

```
    ↪ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↪
    ↪ 0.9 Granulocytes')
35
36 samplesIntegFltGranulocytes <- FinGranulusters(↪
    ↪ samplesIntegFltGranulocytes, resolution = 1.0)
37 DimPlot(samplesIntegFltGranulocytes, reduction = "umap", label = ↪
    ↪ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↪
    ↪ 1.0 Granulocytes')
38
39 clustree(samplesIntegFltGranulocytes)
```

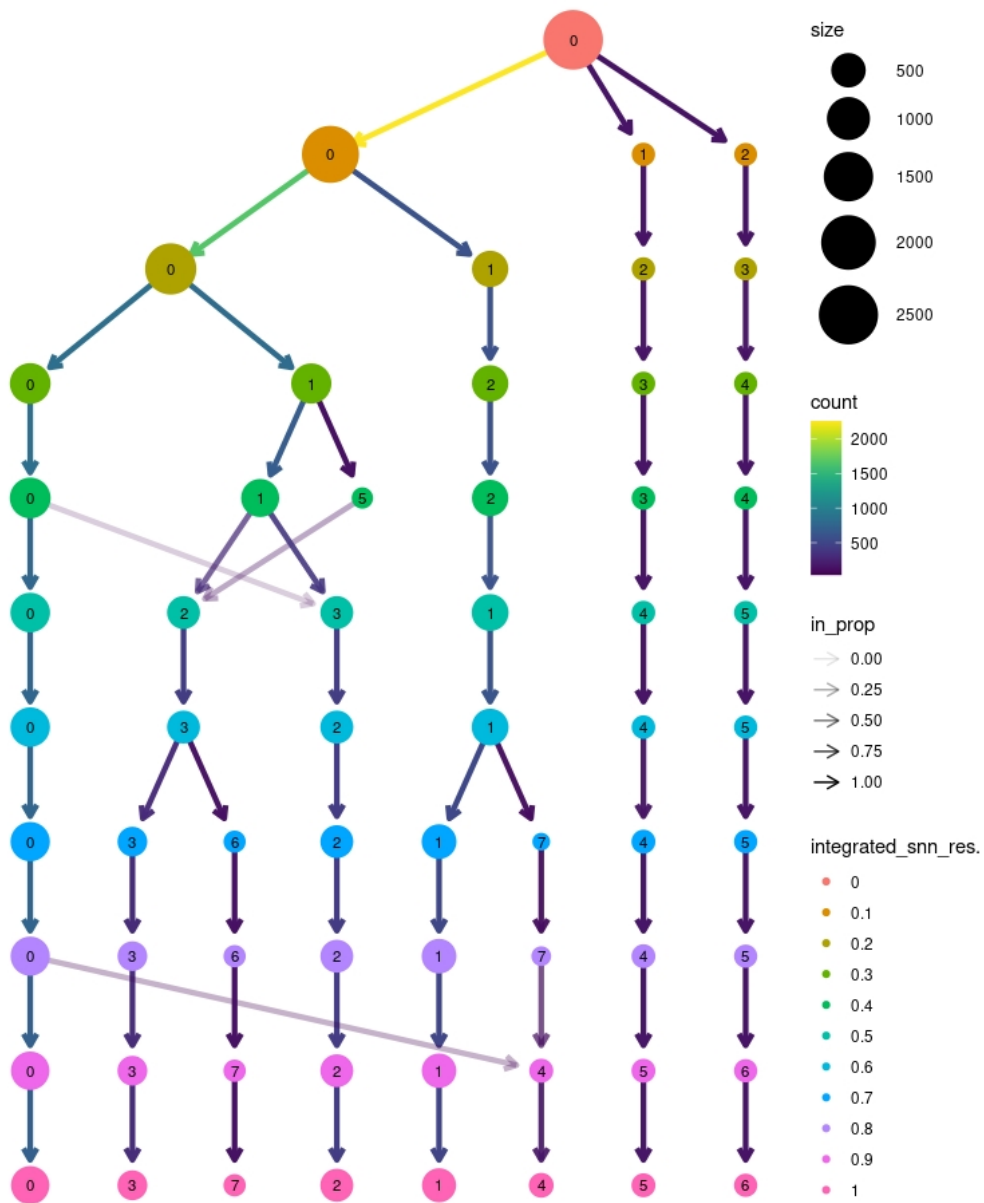


Figure E.4: Granulocytes Clustree

Código E.6: R studio code extract – Granulocytes density plot

```

1 #--- Density plot ---#
2 DefaultAssay(object = samplesIntegFltGranulocytes) <- "SCT"
3
4 samplsList <- SplitObject(samplesIntegFltGranulocytes, split.by = "↔
  ↳ Treatment")
5
6 Granu_Untreated <- samplsList[[1]]
7 Granu_Treated <- samplsList[[2]]

```



```

8
9 saveRDS(Granu_Untreated, "Granu_Untreated.rds")
10 saveRDS(Granu_Treated, "Granu_Treated.rds")
11
12 #- Normal -#
13 p <- DimPlot(samplesIntegFltGranulocytes, group.by = "Treatment", ←
  ↪ reduction = "umap")
14
15 #Prepare data
16 data <- data.frame(samplesIntegFltGranulocytes@meta.data)
17 data$UMAP1 <- Embeddings(samplesIntegFltGranulocytes, "umap")[,1]
18 data$UMAP2 <- Embeddings(samplesIntegFltGranulocytes, "umap")[,2]
19
20 #Create a data frame that doesn't contain a "sample" column.
21 data_bg <- data[, -(which(colnames(data)=="Sample"))]
22
23 #Plot
24 density_plot <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
25   geom_point(colour="#00000000") +
26   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↔
  ↪ =25) +
27   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↔
  ↪ darkolivegreen3", "goldenrod1", "red")) +
28   xlim(c(-10, 10)) + ylim(c(-10, 10)) +
29   #facet_wrap(~Treatment, ncol=3)+
30   theme_classic() + ggtitle("Density") +
31   theme(plot.title = element_text(hjust = 0.5), text = element_↔
  ↪ text(size = 14))
32
33 density_plot
34
35 #- density 2: Untreated -#
36
37 #Prepare data
38 data <- data.frame(Granu_Untreated@meta.data)
39 data$UMAP1 <- Embeddings(Granu_Untreated, "umap")[,1]
40 data$UMAP2 <- Embeddings(Granu_Untreated, "umap")[,2]
41
42 #Create a data frame that doesn't contain a "sample" column.
43 data_bg <- data[, -(which(colnames(data)=="Sample"))]
44
45 #Plot
46 density_plot_2 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
47   geom_point(colour="#00000000") +
48   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↔

```

```

    ↪ =25) +
49 scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
50 xlim(c(-10, 10)) + ylim(c(-12, 10)) +
51 facet_wrap(~Treatment, ncol=3)+
52 theme_classic() + ggtitle("Untreated Density plot") +
53 theme(plot.title = element_text(hjust = 0.5), text = element_↪
    ↪ text(size = 14))
54
55 density_plot_2
56
57 #- density 3: Treated -#
58
59 #Prepare data
60 data <- data.frame(Granu_Treated@meta.data)
61 data$UMAP1 <- Embeddings(Granu_Treated, "umap")[,1]
62 data$UMAP2 <- Embeddings(Granu_Treated, "umap")[,2]
63
64 #Create a data frame that doesn't contain a "sample" column.
65 data_bg <- data[, -(which(colnames(data)=="Sample"))]
66
67 #Plot
68 density_plot_3 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
69   geom_point(colour="#00000000") +
70   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↪
    ↪ =25) +
71   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
72   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
73   facet_wrap(~Treatment, ncol=3)+
74   theme_classic() + ggtitle("Treated Density plot") +
75   theme(plot.title = element_text(hjust = 0.5), text = element_↪
    ↪ text(size = 14))
76
77 density_plot_3
78
79
80 plot_grid(p, density_plot, density_plot_2, density_plot_3, ncol=2)

```

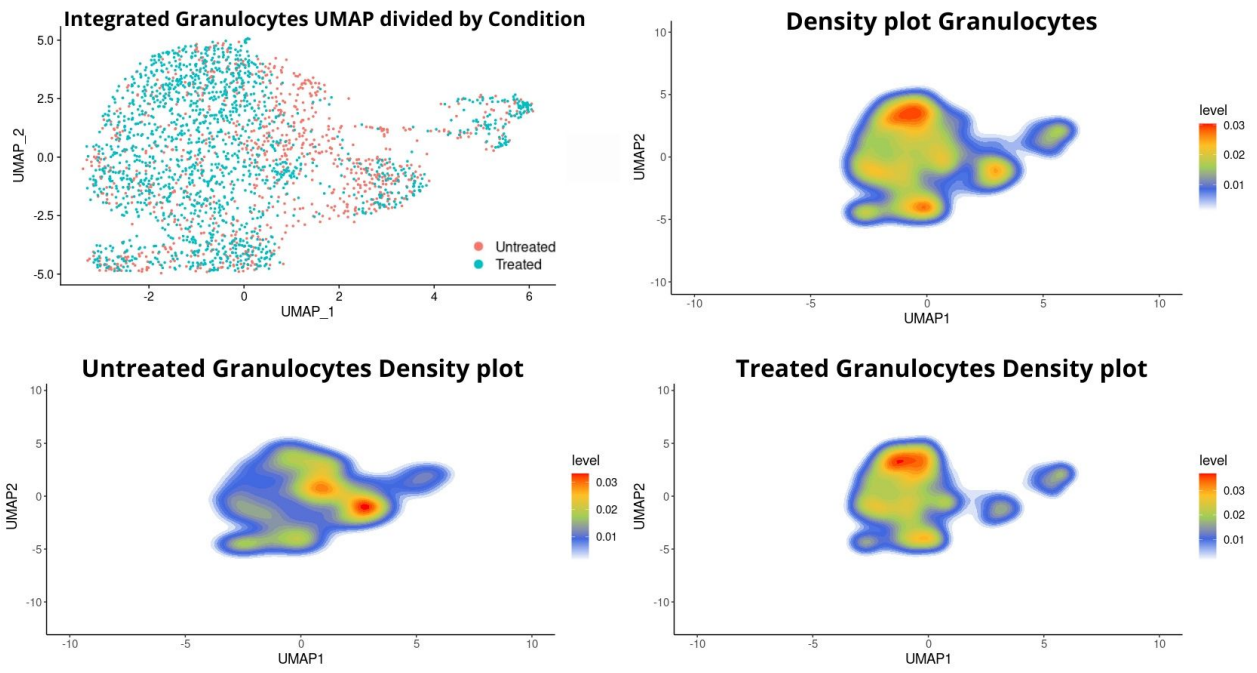


Figure E.5: Granulocytes density plots

Código E.7: R studio code extract – Granulocytes signature analysis

```

1 #--- Signature analysis ---#
2 DefaultAssay(object = samplesIntegFltGranulocytes) <- "integrated"
3
4 #Chosen resolution:
5 samplesIntegFltGranulocytes <- FinGranulusters(↵
  ↵ samplesIntegFltGranulocytes, resolution = 0.1)
6 DimPlot(samplesIntegFltGranulocytes, reduction = "umap", label = ↵
  ↵ FALSE, pt.size = 2, cols = my_colors) + ggtitle('Resolution ↵
  ↵ 0.1 Granulocytes')
7
8 #-- Neutrophil signature --#
9 DefaultAssay(object = samplesIntegFltGranulocytes) <- "SCT"
10
11 Datos_Granu_Zilionis <- read.xlsx("GranuSignatures.xlsx", sheet = "↵
  ↵ Zilionis")
12
13 Genes.mNeutrophils <- Datos_Granu_Zilionis[,1]
14
15 mNeutrophils <- list(Genes.mNeutrophils)
16
17 samplesIntegFltGranulocytes <- AddModuleScore(object = ↵
  ↵ samplesIntegFltGranulocytes, features = mNeutrophils, name = ↵
  ↵ 'mNeutrophils')

```

```
18 FeaturePlot(samplesIntegFltGranulocytes, features = 'mNeutrophils1'↵
↵ , min.cutoff = 0, pt.size = 1.5, cols = c("royalblue", "↵
↵ brown2"))
19 VlnPlot(samplesIntegFltGranulocytes, features = c("mNeutrophils1")↵
↵ , pt.size = 0, cols = my_colors)
```

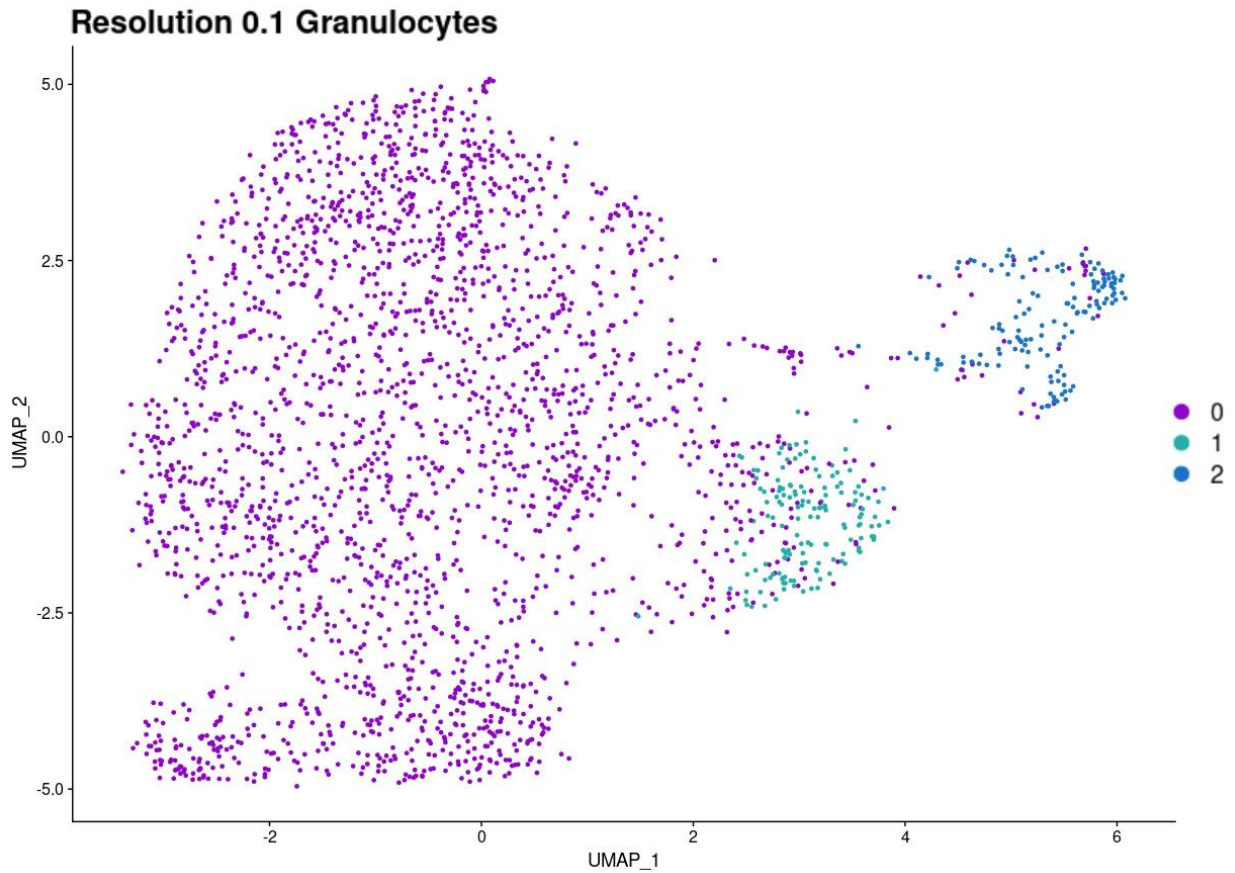


Figure E.6: Integrated Granulocytes UMAP resolution 0.1

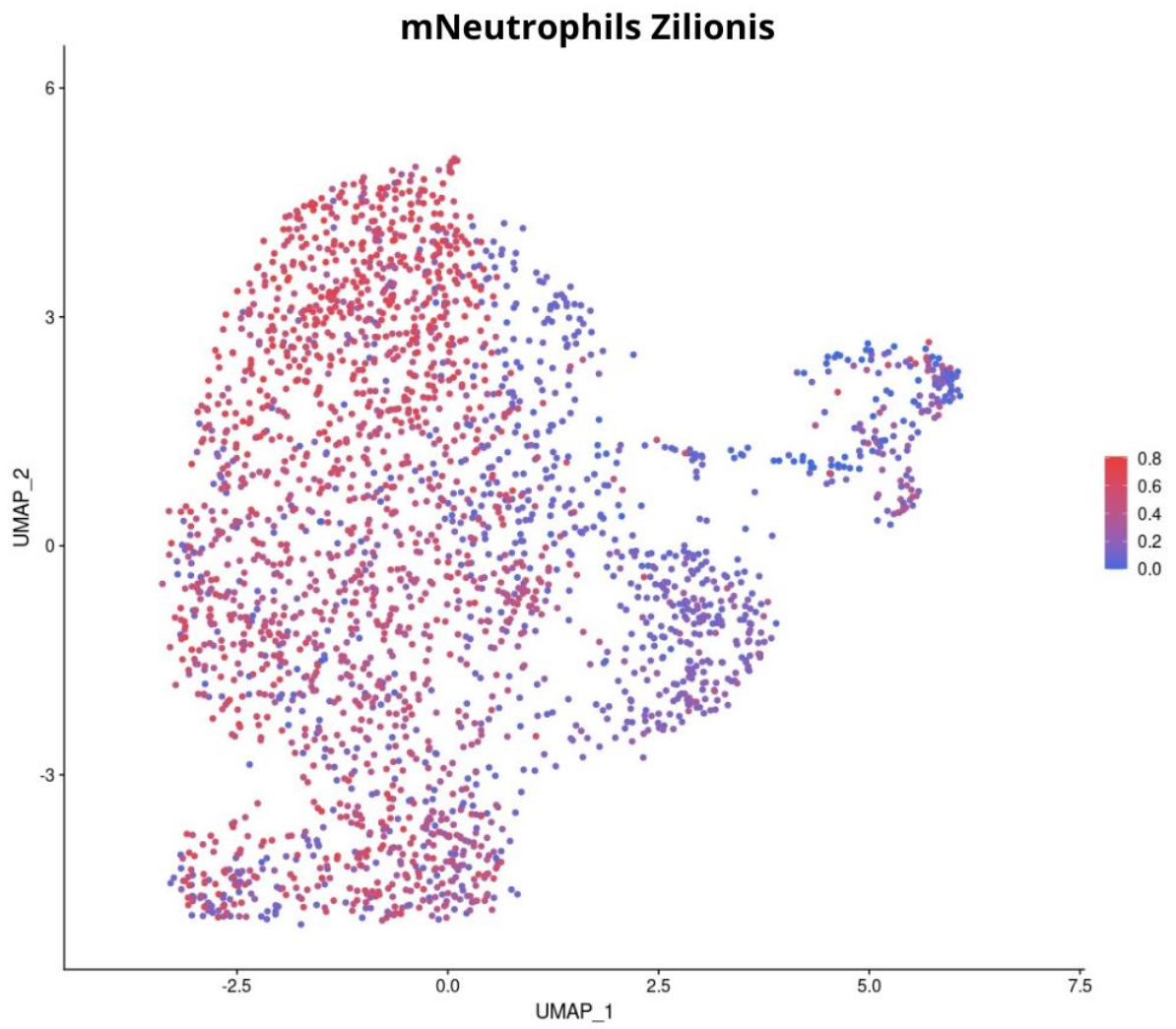


Figure E.7: Zilionis mNeutrophils signature expression level

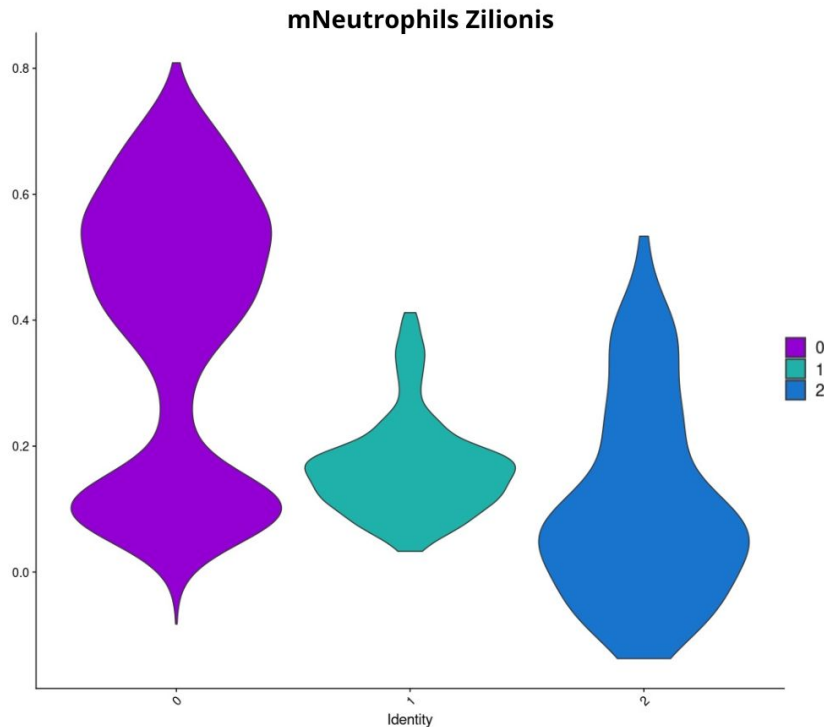


Figure E.8: Zilionis mNeutrophils signature violin plot

Código E.8: R studio code extract – Granulocytes differential gene expression analysis

```

1 #-- Differential gene expression analysis --#
2 DefaultAssay(object = samplesIntegFltGranulocytes) <- "integrated"
3
4 samplesIntegFltGranulocytes.markers <- FindAllMarkers(↵
  ↵ samplesIntegFltGranulocytes, only.pos = TRUE, min.pct = ↵
  ↵ 0.25, logfc.threshold = 0.25)
5 top10Markers <- samplesIntegFltGranulocytes.markers %>% group_by(↵
  ↵ cluster) %>% top_n(n=10, wt=avg_log2FC)
6 top_10 <- unique(top10Markers$gene)
7 DotPlot(samplesIntegFltGranulocytes, features = top_10, cols=c("↵
  ↵ royalblue", "brown2")) + RotatedAxis() + ggtitle('↵
  ↵ Resolution 0.1 markers')
```

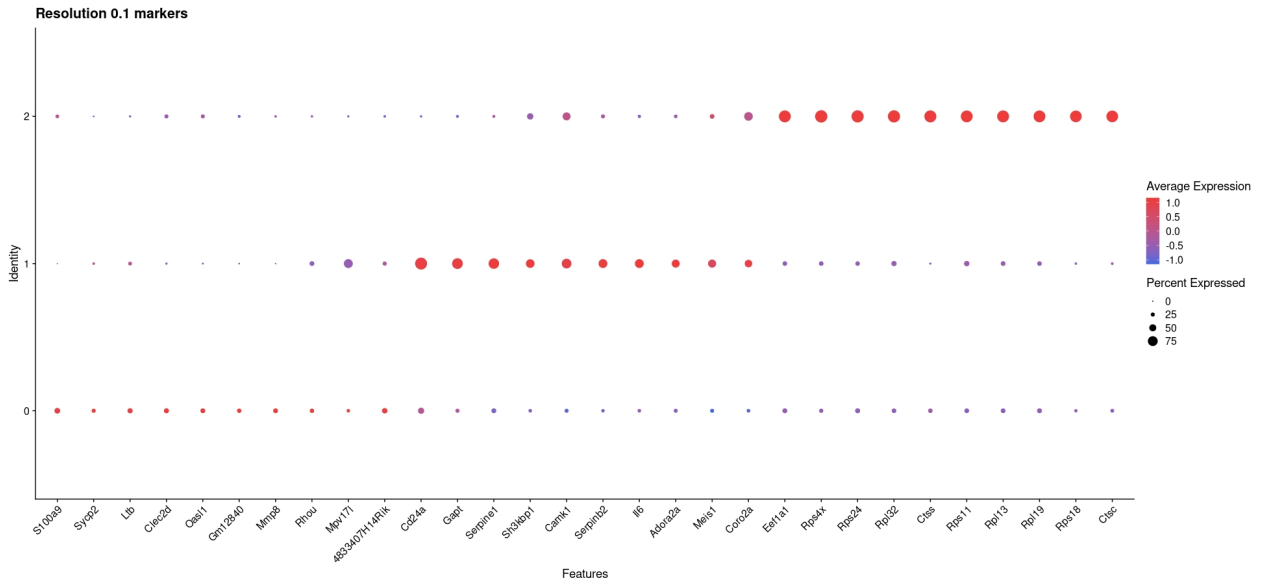


Figure E.9: Granulocytes resolution 0.1 Dotplot

Código E.9: R studio code extract – Granulocytes Translationally active
Neutrophiles signature analysis

```

1
2 #- Alshetaiwi Monocle States -#
3 DefaultAssay(object = samplesIntegFltGranulocytes) <- "SCT"
4 Datos_Granu_Alshetaiwi <- read.xlsx("GranuSignatures.xlsx", sheet = <-
  <- "AlshetaiwiMonocle")
5
6 Genes.TranslationallyAct <- Datos_Granu_Alshetaiwi[,5]
7
8 TranslationallyAct <- list(Genes.TranslationallyAct)
9
10 samplesIntegFltGranulocytes <- AddModuleScore(object = <-
  <- samplesIntegFltGranulocytes, features = TranslationallyAct, <-
  <- name = 'TranslationallyAct')
11 FeaturePlot(samplesIntegFltGranulocytes, features = '<-
  <- TranslationallyAct1', min.cutoff = 0, pt.size = 1.5, cols = <-
  <- c("royalblue", "brown2"))
12 VlnPlot(samplesIntegFltGranulocytes, features = c("<-
  <- TranslationallyAct1"), pt.size = 0, cols = my_colors)

```


Translationally active Neutrophils Alshetaiwi

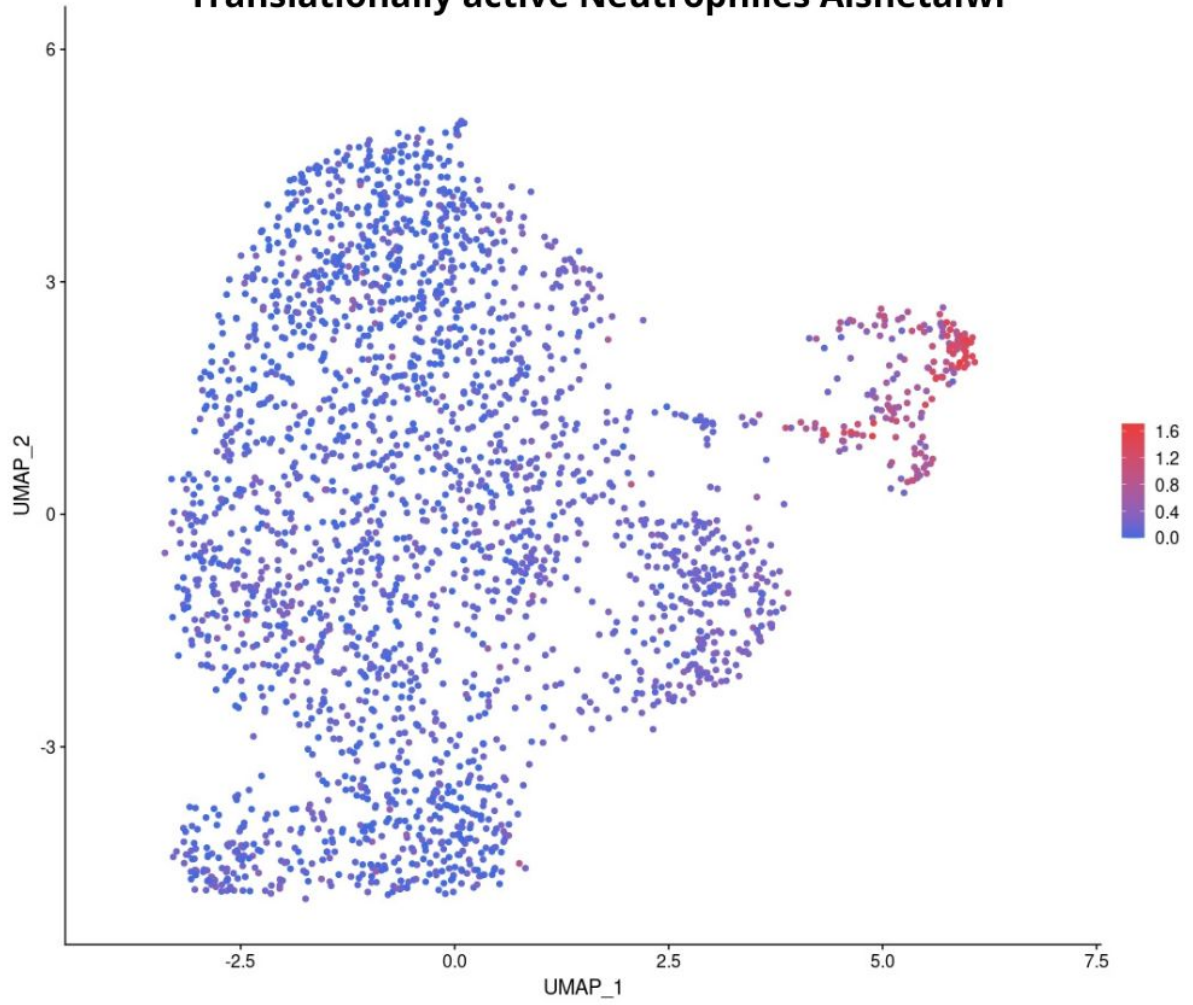


Figure E.10: Alshetaiwi Translationally active Neutrophils signature expression level

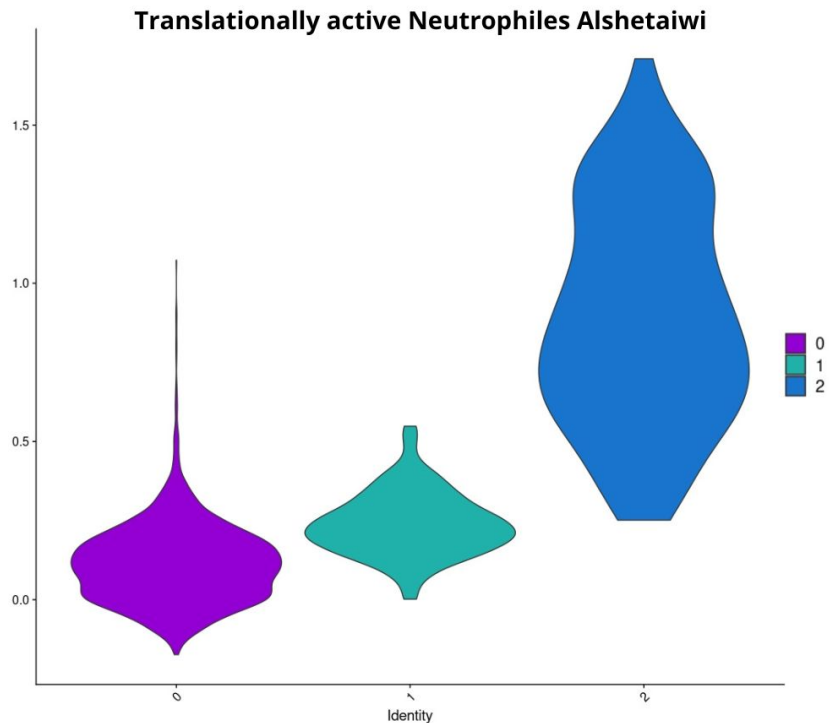


Figure E.11: Alshetaiwi Translationally active Neutrophils signature violin plot

Código E.10: R studio code extract – Granulocytes Eosinophils key gene signature expression analysis

```

1 Eosinophils_Gurtner <- list(c("Siglecf", "Il5ra", "Ccr3", "Epx"))
2 samplesIntegFltGranulocytes <- AddModuleScore(object = ↵
  ↵ samplesIntegFltGranulocytes, features = Eosinophils_Gurtner, ↵
  ↵ name = 'Eosinophils_Gurtner',)
3 FeaturePlot(samplesIntegFltGranulocytes, features = 'Eosinophils_↵
  ↵ Gurtner1', min.cutoff = 0, pt.size = 2, cols = c("royalblue"↵
  ↵ , "brown2"))
4 VlnPlot(samplesIntegFltGranulocytes, features = c("Eosinophils_↵
  ↵ Gurtner1"), pt.size = 0, cols = my_colors)

```

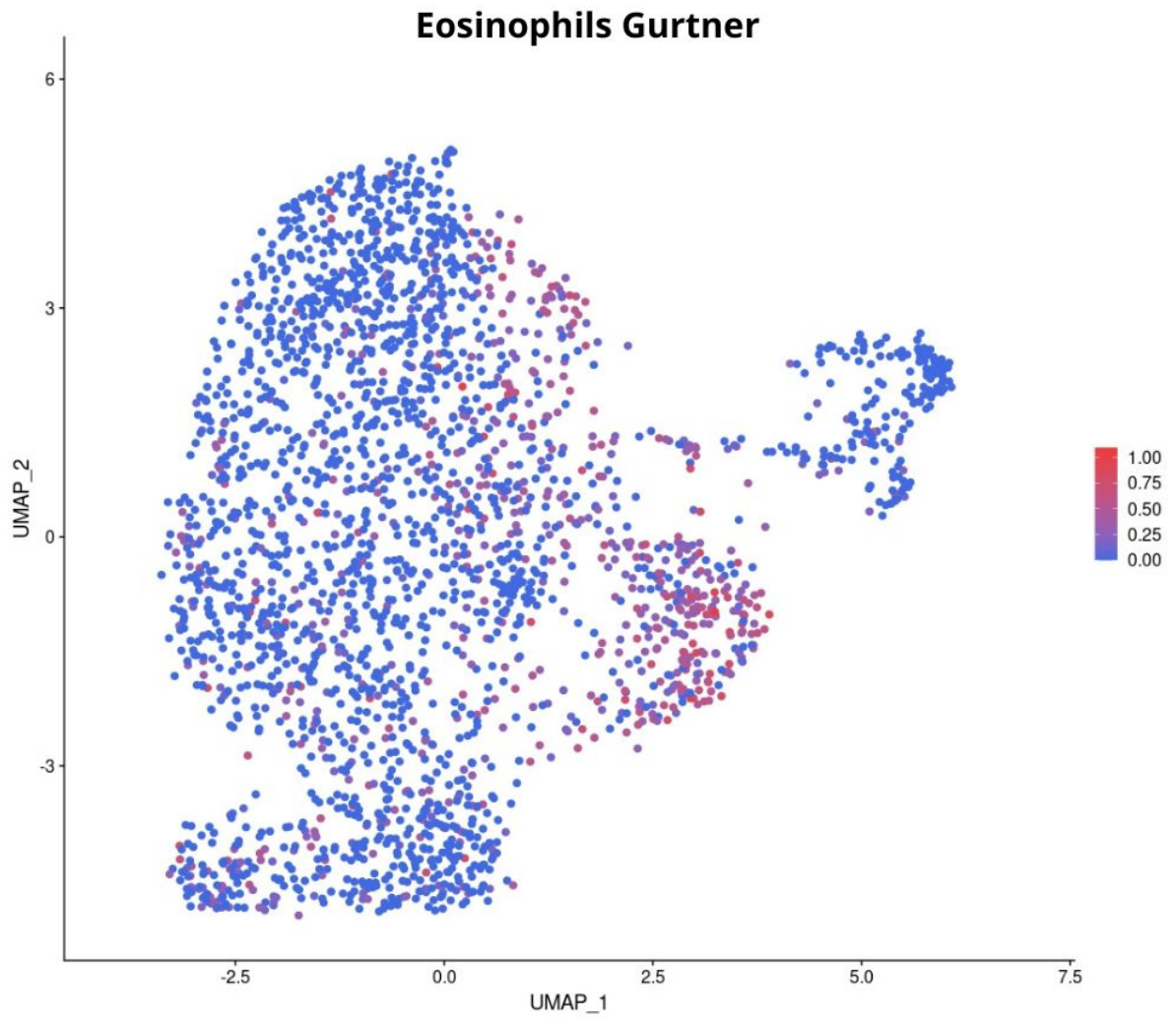


Figure E.12: Gurtner Eosinophils key genes signature expression level

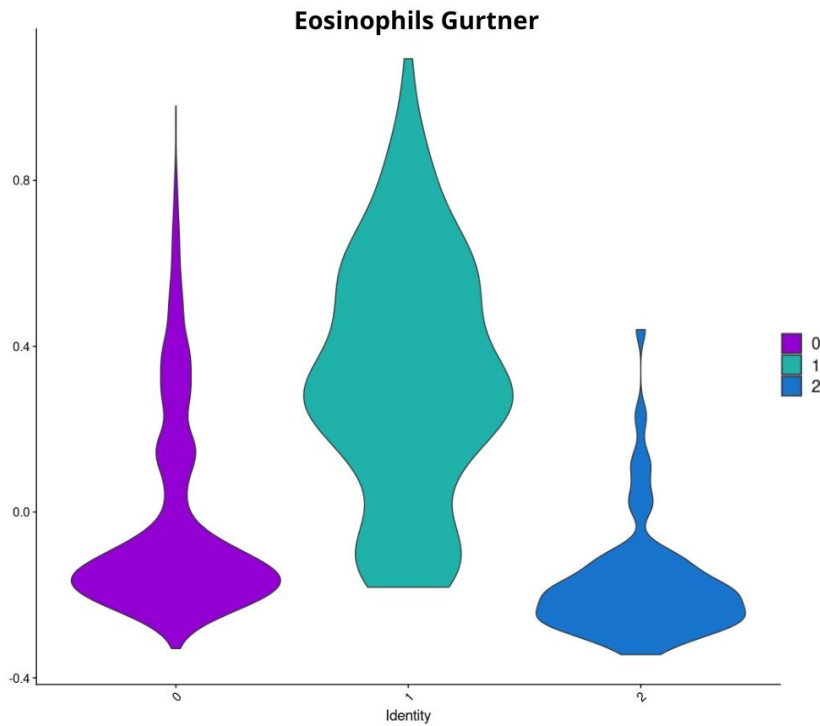


Figure E.13: Gurtner Eosinophils key genes signature violin plot

Código E.11: R studio code extract – Granulocytes subpopulation division

```

1 #-- New classification --#
2 samplesIntegFltGranulocytes <- RenameIdents(object = ←
  ↪ samplesIntegFltGranulocytes, 0 = Neutrophils, 1 = ←
  ↪ Eosinophils, 2 = Translationally active Neutrophiles)
3
4 DimPlot(samplesIntegFltGranulocytes, reduction = "umap", label = ←
  ↪ FALSE, pt.size = 1.5, cols = my_colors, label.size = 6) + ←
  ↪ ggtitle('Granulocytes subpopulations')
```

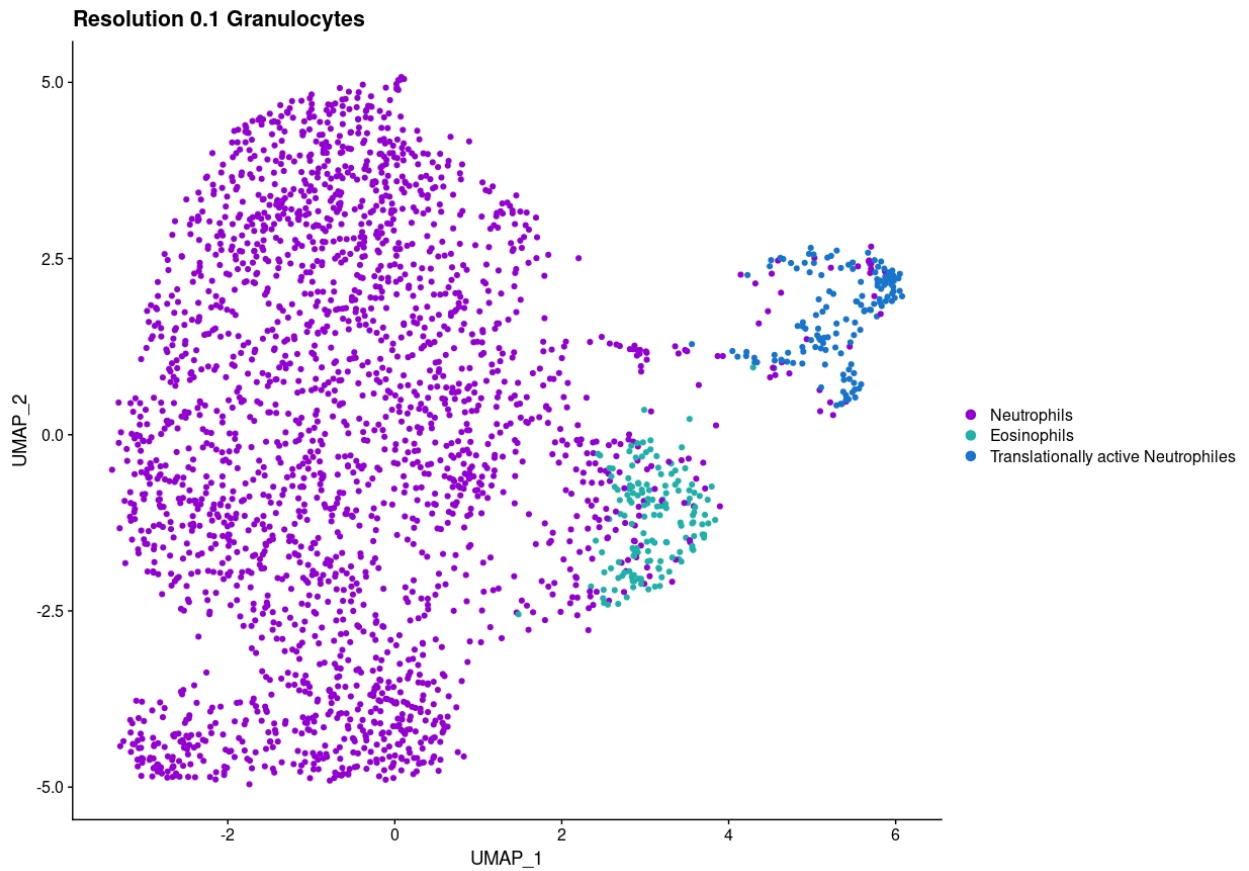


Figure E.14: Granulocytes subpopulations

Código E.12: R studio code extract – Granulocytes Barplots

```

1
2 samplesIntegFltGranulocytes[["Subpopulation"]] <- <-
  <- samplesIntegFltGranulocytes@meta.data$integrated_snn_res.0.1
3 dittoBarPlot(object = samplesIntegFltGranulocytes, scale = "<-
  <- percent", var = "Subpopulation", group.by = "Treatment", <-
  <- color.panel = my_colors, x.labels = c("Untreated", "Treated"<-
  <- ), xlab = "Condition", main = "")

```

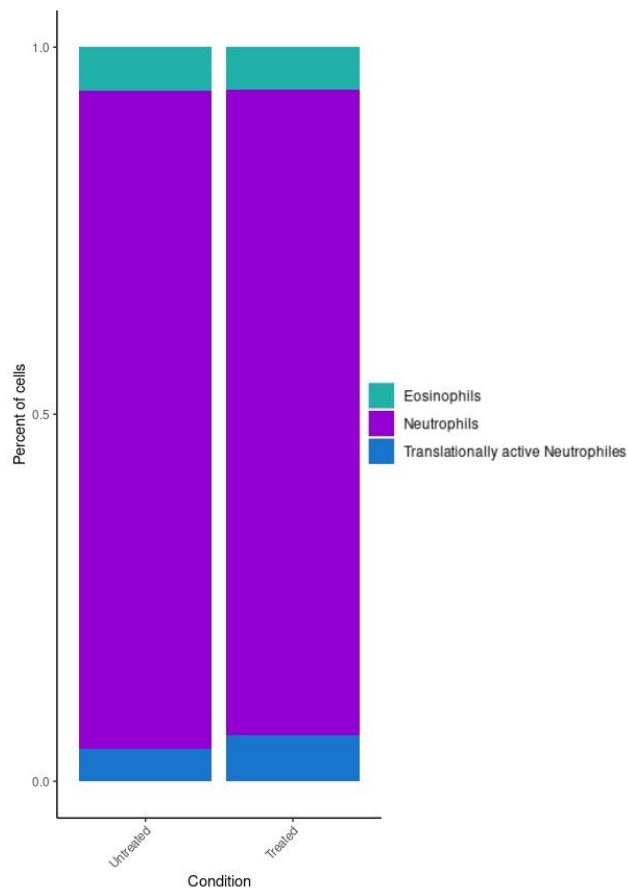


Figure E.15: Barplot of Granulocytes subpopulations percentage by Conditions

Código E.13: R studio code extract – Neutrophil subset re-integration

```

1 samplesIntegFltNeutrophils <- subset(samplesIntegFltGranulocytes, ←
  ↪ idents=c(Neutrophils), invert=FALSE)
2 DimPlot(samplesIntegFltNeutrophils, reduction = "umap", label = ←
  ↪ TRUE, pt.size = 1, cols = my_colors)
3
4 samplesIntegrated_Neutrophils.list <- SplitObject(←
  ↪ samplesIntegFltNeutrophils, split.by = "orig.ident")
5
6 Untreated1 <- samplesIntegrated_Neutrophils.list[[1]]
7 Untreated2 <- samplesIntegrated_Neutrophils.list[[2]]
8 Treated1 <- samplesIntegrated_Neutrophils.list[[3]]
9 Treated2 <- samplesIntegrated_Neutrophils.list[[4]]
10
11 samplesNeutrophils.list <- list(Untreated1, Untreated2, Treated1, ←
  ↪ Treated2)
12

```

```

13 samplesNeutrophils.list <- lapply(X= samplesNeutrophils.list , FUN <-
    <- = function(x){
14   x <- SCTransform(object = x, vars.to.regress = "percent.mt", <-
    <- verbose = FALSE, variable.features.n = 2000)})
15
16 #Features
17 samplesNeutrophils.features <- SelectIntegrationFeatures(object.<-
    <- list = samplesNeutrophils.list )
18 samplesNeutrophils.list <- PrepSCTIntegration(object.list = <-
    <- samplesNeutrophils.list , anchor.features = <-
    <- samplesNeutrophils.features)
19
20 #Anchors
21 samplesNeutrophils.anchors <- FindIntegrationAnchors(object.list =<-
    <- samplesNeutrophils.list , normalization.method = "SCT", <-
    <- anchor.features = samplesNeutrophils.features)
22 save(samplesNeutrophils.anchors , file=file.path("~/", "<-
    <- samplesNeutrophils.anchors.rdata"))
23
24 #Integration
25 samplesIntegFltNeutrophils <- IntegrateData(anchorset = <-
    <- samplesNeutrophils.anchors , normalization.method = "SCT")
26 saveRDS(samplesIntegFltNeutrophils , "samplesIntegFltNeutrophils.rds<-
    <- ")

```

Código E.14: R studio code extract – Neutrophils linear reduction

```

1 #- PCA reduction -#
2 samplesIntegFltNeutrophils <- RunPCA(samplesIntegFltNeutrophils , <-
    <- npcs = 50, verbose=F)
3
4 #Elbow plot
5 ElbowPlot(samplesIntegFltNeutrophils , ndims = 50, reduction = "pca<-
    <- ")

```

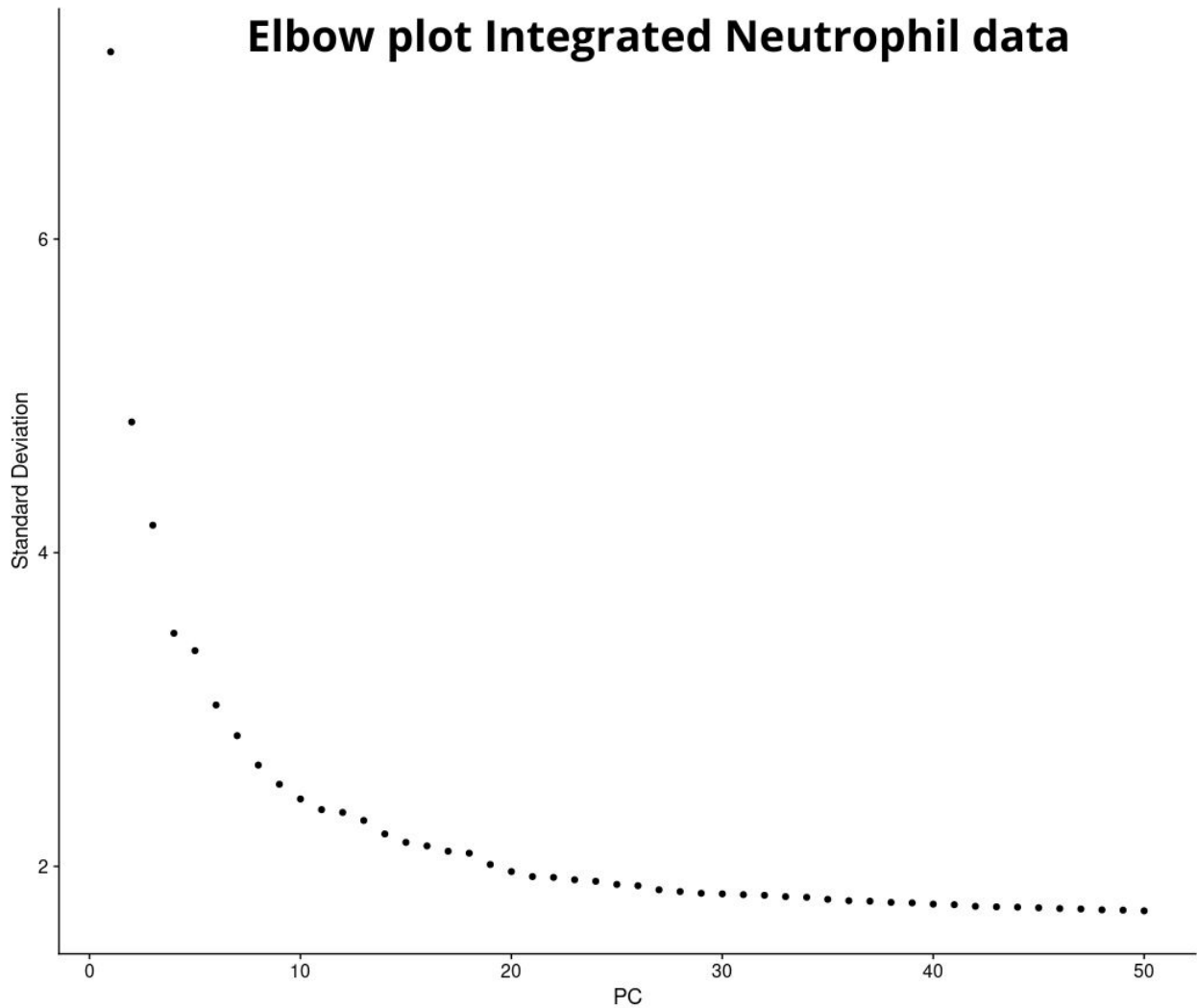


Figure E.16: Elbow plot Neutrophils

Código E.15: R studio code extract – Neutrophils non-linear reduction

```

1 #- UMAP Reduction -#
2 samplesIntegFltNeutrophils <- RunUMAP(samplesIntegFltNeutrophils , <-
  ↪ dims=1:30, verbose=F)
3
4 #UMAP by Condition
5 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , group.by =<-
  ↪ 'Treatment' , pt.size = 1.5) + ggtitle("Integrated ↪
  ↪ Neutrophils UMAP divided by Condition") + scale_color_↪
  ↪ manual(labels = c("Untreated" , "Treated") , values = c("#↪
  ↪ F8766D" , "#00bfc4"))
6
7 #UMAP by Sample
8 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , group.by =<-

```

```
↪ 'orig.ident', pt.size = 1.5) + ggtitle("Integrated ↪  
↪ Neutrophils UMAP divided by Sample") + scale_color_manual(↪  
↪ labels = c("Untreated1", "Untreated2", "Treated1", "Treated2" ↪  
↪ ), values = my_colors)
```

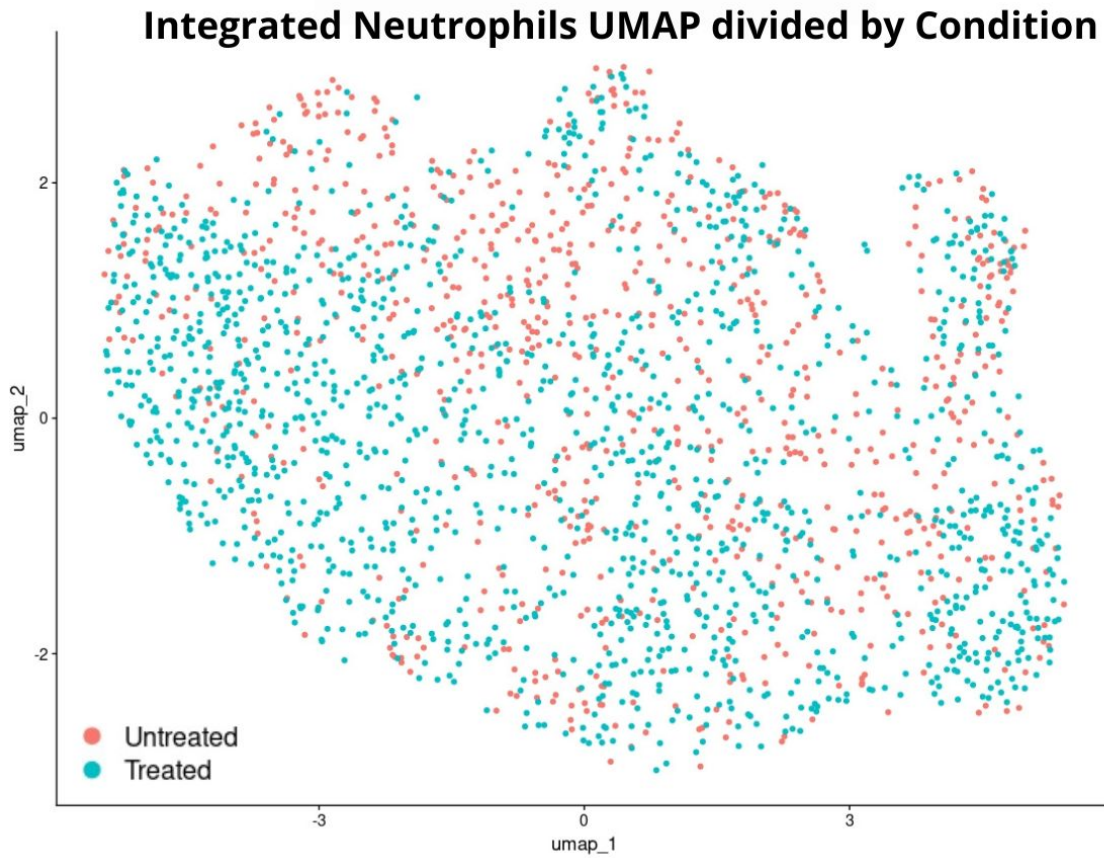


Figure E.17: UMAP of Neutrophils divided by Condition

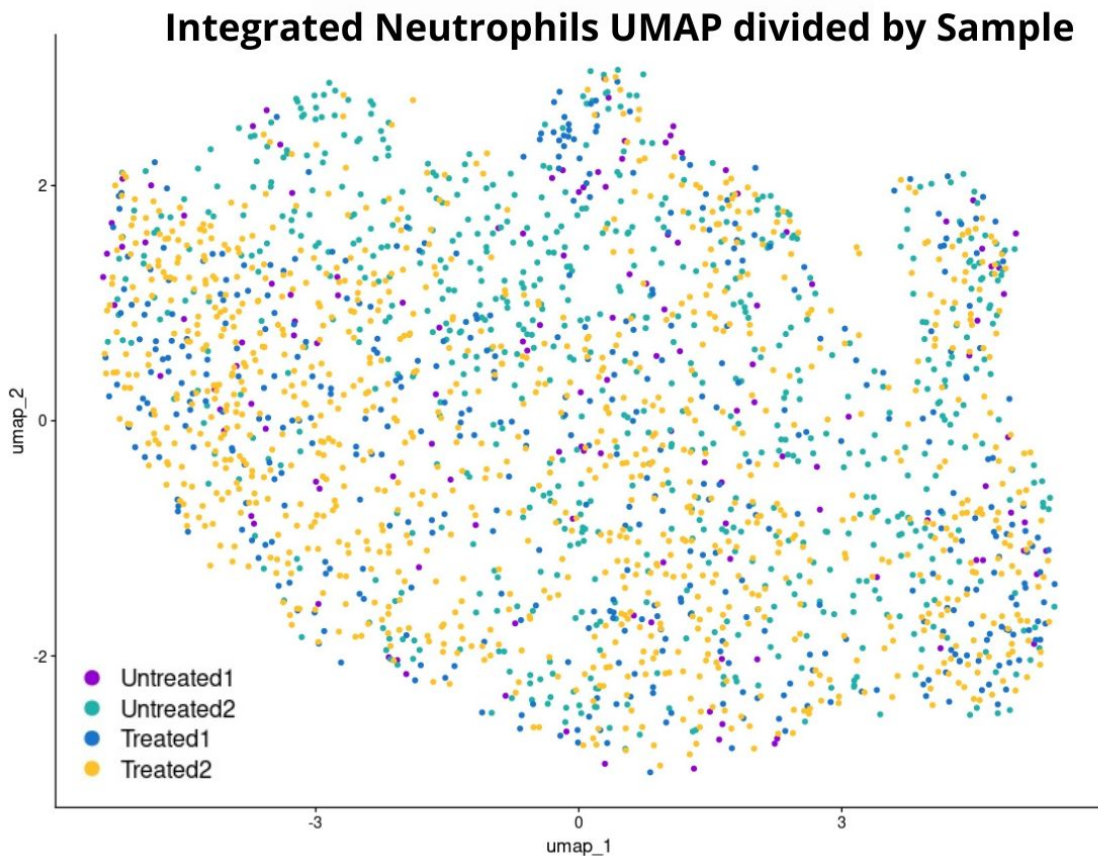


Figure E.18: UMAP of Neutrophils divided by Sample

Código E.16: R studio code extract – Neutrophils clustering

```

1 #--- Clustering ---#
2 DefaultAssay(object = samplesIntegFltNeutrophils) <- "integrated"
3
4 samplesIntegFltNeutrophils <- FindNeighbors(↵
  ↵ samplesIntegFltNeutrophils, dims = 1:30, verbose = FALSE)
5
6 samplesIntegFltNeutrophils <- FindClusters(↵
  ↵ samplesIntegFltNeutrophils, resolution = 0)
7 DimPlot(samplesIntegFltNeutrophils, reduction = "umap", label = ↵
  ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution 0↵
  ↵ Neutrophils')
8
9 samplesIntegFltNeutrophils <- FindClusters(↵
  ↵ samplesIntegFltNeutrophils, resolution = 0.1)
10 DimPlot(samplesIntegFltNeutrophils, reduction = "umap", label = ↵
  ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
  ↵ 0.1 Neutrophils')
11

```

```

12 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.2)
13 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.2 Neutrophils')
14
15 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.3)
16 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.3 Neutrophils')
17
18 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.4)
19 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.4 Neutrophils')
20
21 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.5)
22 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.5 Neutrophils')
23
24 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.6)
25 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.6 Neutrophils')
26
27 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.7)
28 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.7 Neutrophils')
29
30 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.8)
31 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
    ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
    ↵ 0.8 Neutrophils')
32
33 samplesIntegFltNeutrophils <- FindClusters(↵
    ↵ samplesIntegFltNeutrophils , resolution = 0.9)
34 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵

```

```
    ↪ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↪
    ↪ 0.9 Neutrophils')
35
36 samplesIntegFltNeutrophils <- FindClusters(↪
    ↪ samplesIntegFltNeutrophils, resolution = 1.0)
37 DimPlot(samplesIntegFltNeutrophils, reduction = "umap", label = ↪
    ↪ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↪
    ↪ 1.0 Neutrophils')
38
39 clustree(samplesIntegFltNeutrophils)
```

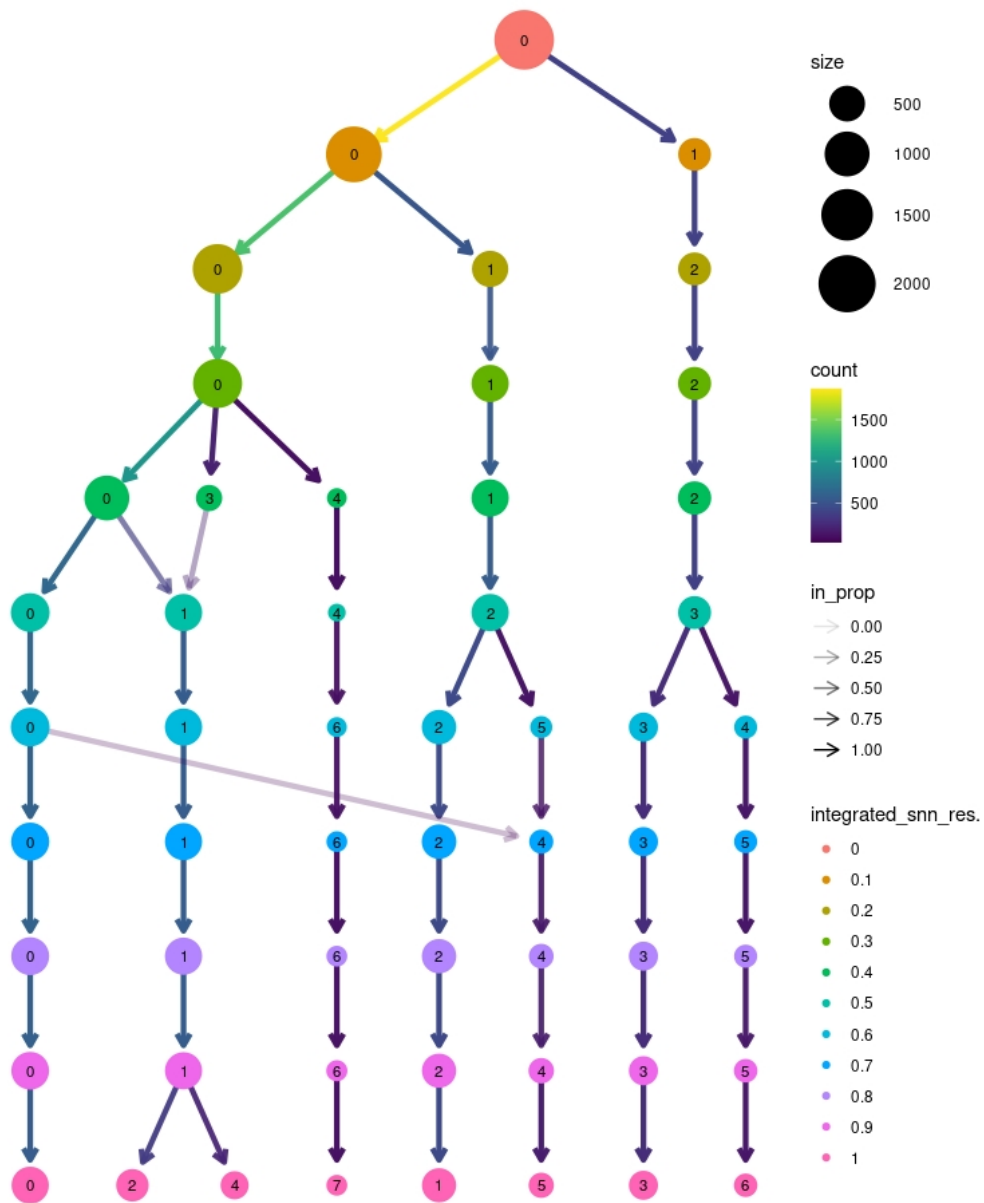


Figure E.19: Neutrophils Clustree

Código E.17: R studio code extract – Neutrophils density plot

```

1 #--- Density plot ---#
2 DefaultAssay(object = samplesIntegFltGranulocytes) <- "SCT"
3
4 samplsList <- SplitObject(samplesIntegFltGranulocytes, split.by = "↔
  ↳ Treatment")
5
6 Granu_Untreated <- samplsList[[1]]
7 Granu_Treated <- samplsList[[2]]

```

```

8
9 saveRDS(Granu_Untreated, "Granu_Untreated.rds")
10 saveRDS(Granu_Treated, "Granu_Treated.rds")
11
12 #- Normal -#
13 p <- DimPlot(samplesIntegFltGranulocytes, group.by = "Treatment", ←
  ↪ reduction = "umap")
14
15 #Prepare data
16 data <- data.frame(samplesIntegFltGranulocytes@meta.data)
17 data$UMAP1 <- Embeddings(samplesIntegFltGranulocytes, "umap")[,1]
18 data$UMAP2 <- Embeddings(samplesIntegFltGranulocytes, "umap")[,2]
19
20 #Create a data frame that doesn't contain a "sample" column.
21 data_bg <- data[, -(which(colnames(data)=="Sample"))]
22
23 #Plot
24 density_plot <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
25   geom_point(colour="#00000000") +
26   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↔
  ↪ =25) +
27   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↔
  ↪ darkolivegreen3", "goldenrod1", "red")) +
28   xlim(c(-10, 10)) + ylim(c(-10, 10)) +
29   #facet_wrap(~Treatment, ncol=3)+
30   theme_classic() + ggtitle("Density") +
31   theme(plot.title = element_text(hjust = 0.5), text = element_↔
  ↪ text(size = 14))
32
33 density_plot
34
35 #- density 2: Untreated -#
36
37 #Prepare data
38 data <- data.frame(Granu_Untreated@meta.data)
39 data$UMAP1 <- Embeddings(Granu_Untreated, "umap")[,1]
40 data$UMAP2 <- Embeddings(Granu_Untreated, "umap")[,2]
41
42 #Create a data frame that doesn't contain a "sample" column.
43 data_bg <- data[, -(which(colnames(data)=="Sample"))]
44
45 #Plot
46 density_plot_2 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
47   geom_point(colour="#00000000") +
48   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↔

```

```

    ↪ =25) +
49 scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
50 xlim(c(-10, 10)) + ylim(c(-12, 10)) +
51 facet_wrap(~Treatment, ncol=3)+
52 theme_classic() + ggtitle("Untreated Density plot") +
53 theme(plot.title = element_text(hjust = 0.5), text = element_↪
    ↪ text(size = 14))
54
55 density_plot_2
56
57 #- density 3: Treated -#
58
59 #Prepare data
60 data <- data.frame(Granu_Treated@meta.data)
61 data$UMAP1 <- Embeddings(Granu_Treated, "umap")[,1]
62 data$UMAP2 <- Embeddings(Granu_Treated, "umap")[,2]
63
64 #Create a data frame that doesn't contain a "sample" column.
65 data_bg <- data[, -(which(colnames(data)=="Sample"))]
66
67 #Plot
68 density_plot_3 <- ggplot(data, aes(x=UMAP1, y=UMAP2)) +
69   geom_point(colour="#00000000") +
70   stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins↪
    ↪ =25) +
71   scale_fill_gradientn(colors = c("#4169E100", "royalblue", "↪
    ↪ darkolivegreen3", "goldenrod1", "red")) +
72   xlim(c(-10, 10)) + ylim(c(-12, 10)) +
73   facet_wrap(~Treatment, ncol=3)+
74   theme_classic() + ggtitle("Treated Density plot") +
75   theme(plot.title = element_text(hjust = 0.5), text = element_↪
    ↪ text(size = 14))
76
77 density_plot_3
78
79
80 plot_grid(p, density_plot, density_plot_2, density_plot_3, ncol=2)

```

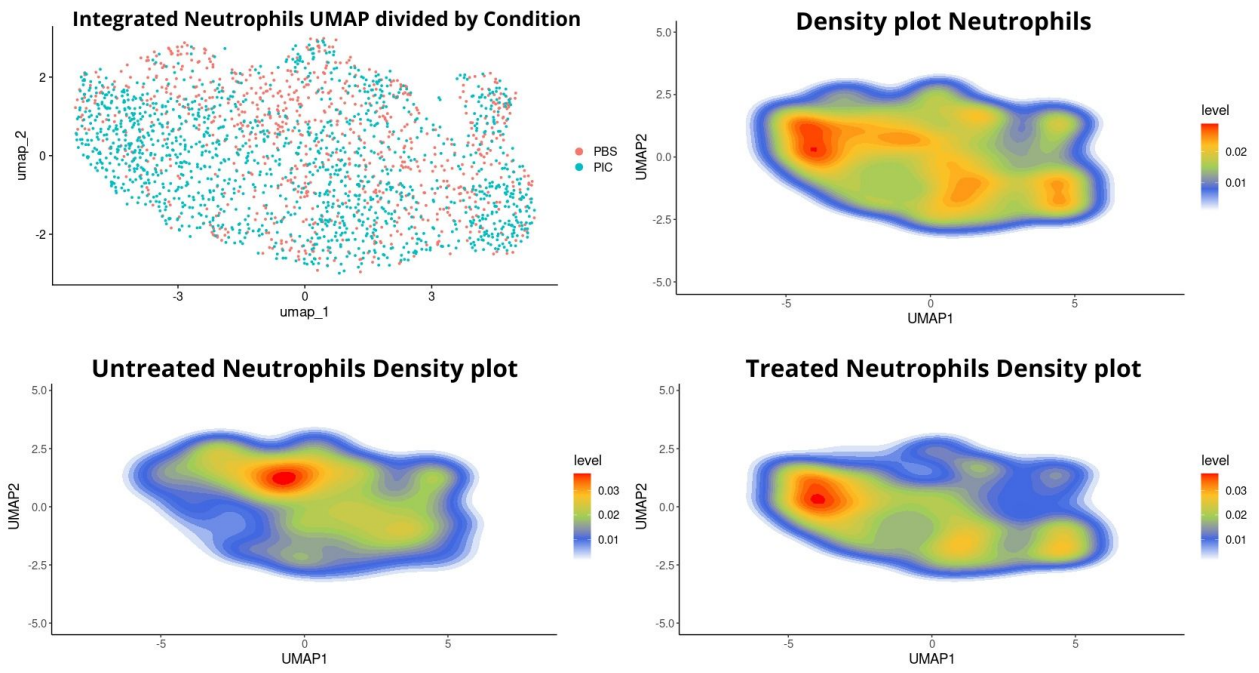


Figure E.20: Neutrophils density plots

Código E.18: R studio code extract – Chosen resolution for Neutrophil analysis

```

1
2 #- Chosen resolution -#
3 samplesIntegFltNeutrophils <- FindClusters(↵
  ↵ samplesIntegFltNeutrophils , resolution = 0.1)
4 DimPlot(samplesIntegFltNeutrophils , reduction = "umap" , label = ↵
  ↵ FALSE, pt.size = 2, cols = my_colors) + ggtitle('Resolution ↵
  ↵ 0.7 Neutrophils')
```

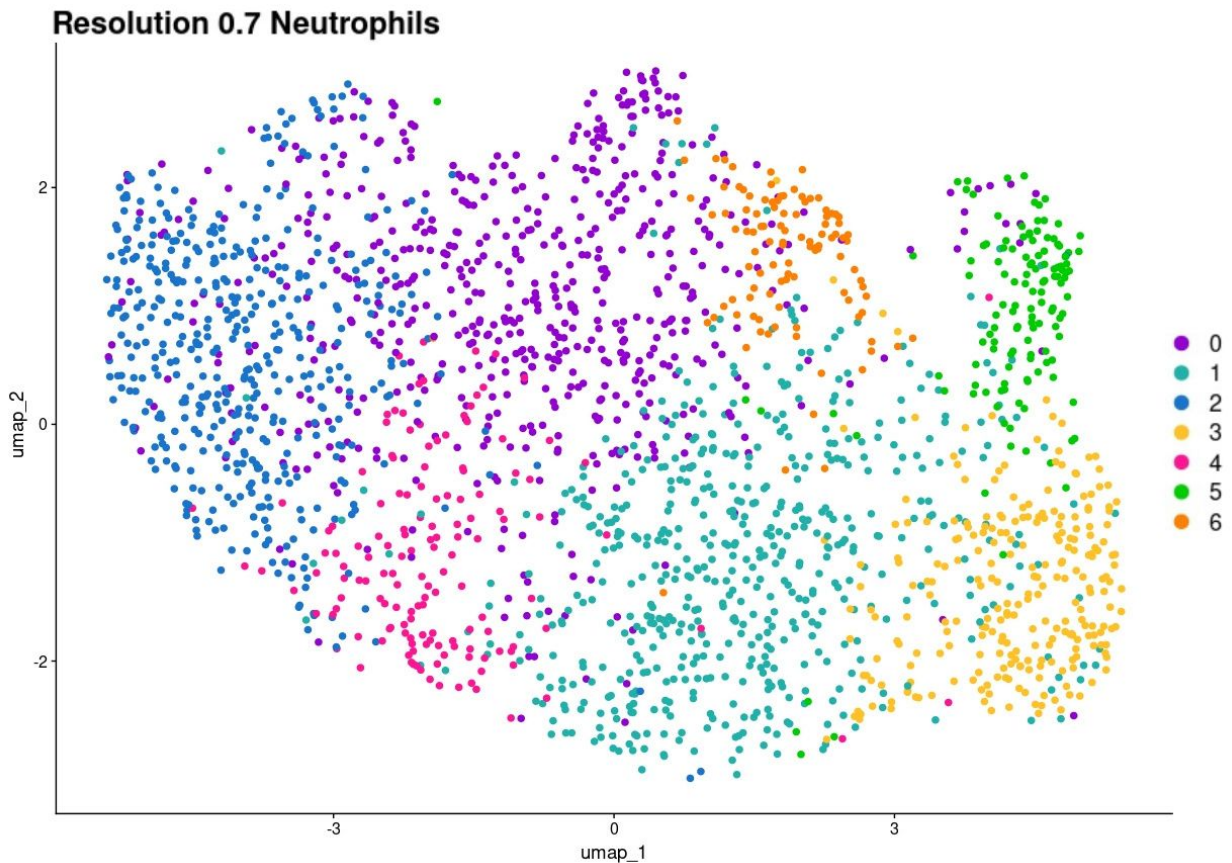



Figure E.21: Integrated Neutrophils UMAP resolution 0.7

Código E.19: R studio code extract – Neutrophils trajectory analysis using Monocle2 part 1

```

1 samplesIntegFltNeutrophils <- FindClusters(↵
  ↵ samplesIntegFltNeutrophils, resolution = 0.7)
2 DimPlot(samplesIntegFltNeutrophils, reduction = "umap", label = ↵
  ↵ TRUE, pt.size = 1, cols = my_colors) + ggtitle('Resolution ↵
  ↵ 0.7 Neutrophils')
3
4 data <- as(as.matrix(samplesIntegFltNeutrophils@assays$SCT@data), ↵
  ↵ 'sparseMatrix')
5
6 pd <- new('AnnotatedDataFrame', data = ↵
  ↵ samplesIntegFltNeutrophils@meta.data)
7
8 fData <- data.frame(gene_short_name = row.names(data), row.names =↵
  ↵ row.names(data))
9 fd <- new('AnnotatedDataFrame', data = fData)
10
11 #Construct monocle cds

```



```

12 monocle_cds <- newCellDataSet(data,
13                               phenoData = pd,
14                               featureData = fd,
15                               lowerDetectionLimit = 0.5,
16                               expressionFamily = negbinomial.size<-
    ↪ ())
17
18 monocle_cds <- estimateSizeFactors(monocle_cds)
19 monocle_cds <- estimateDispersions(monocle_cds)
20
21 expressed_genes <- row.names(samplesIntegFltNeutrophils)
22 diff_test_res <- differentialGeneTest(monocle_cds[expressed_genes<-
    ↪ ],,
23                                       fullModelFormulaStr = "~<-
    ↪ seurat_clusters")
24 ordering_genes <- row.names(subset(diff_test_res, qval < 0.01))
25
26 monocle_cds <- setOrderingFilter(monocle_cds, ordering_genes)
27 plot_ordering_genes(monocle_cds)
28
29 monocle_cds <- reduceDimension(monocle_cds, max_components = 2,
30                               method = 'DDRTree')
31
32 monocle_cds <- orderCells(monocle_cds)
33
34 #- Monocle State graphic trajectory -#
35 plot_cell_trajectory(monocle_cds, color_by = "State")
36
37 #- Monocle Pseudotime graphic trajectory -#
38 plot_cell_trajectory(monocle_cds, color_by = "Pseudotime")
39
40 #- Monocle graphic trajectory using Seurat Clusters -#
41 plot_cell_trajectory(monocle_cds, color_by = "seurat_clusters") + <-
    ↪ scale_color_manual(values=my_colors)
42
43 plot_cell_trajectory(monocle_cds, color_by = "seurat_clusters") + <-
    ↪ facet_wrap(~seurat_clusters, nrow = 1) + scale_color_manual(<-
    ↪ values=my_colors)

```

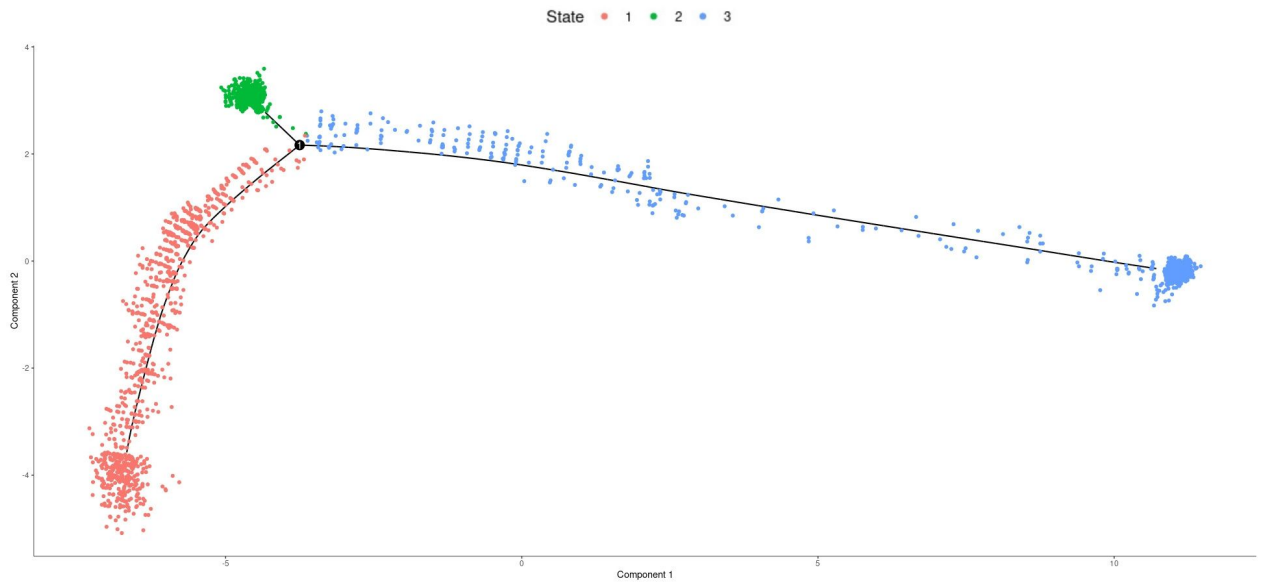


Figure E.22: Monocle2 Neutrophils trajectory colored by State

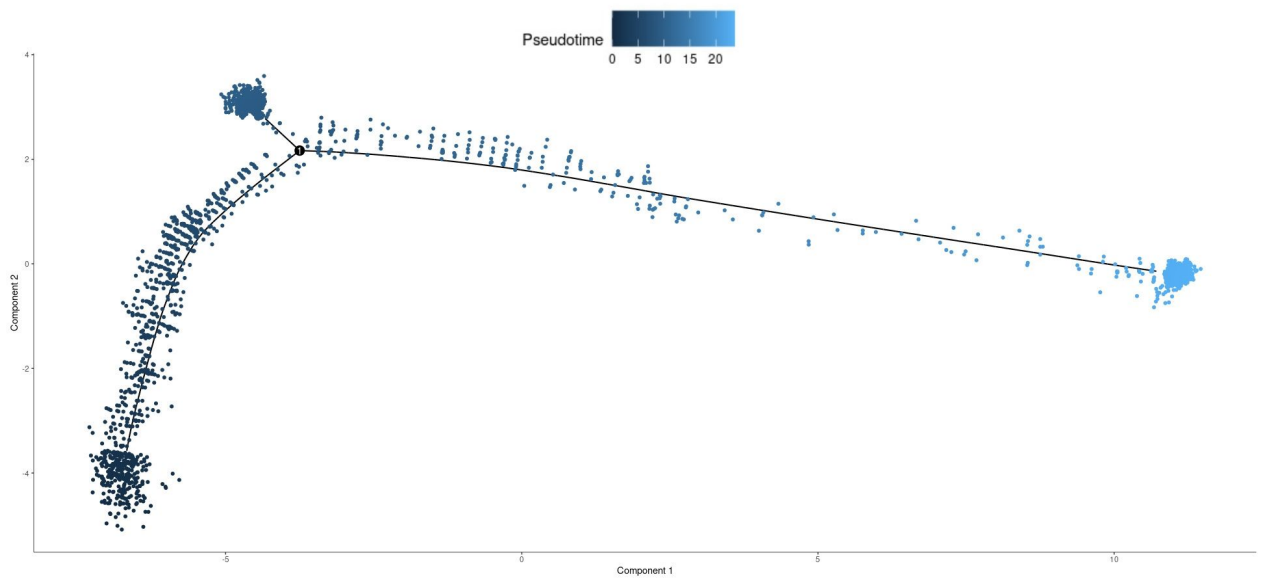


Figure E.23: Monocle2 Neutrophils trajectory colored by Pseudotime

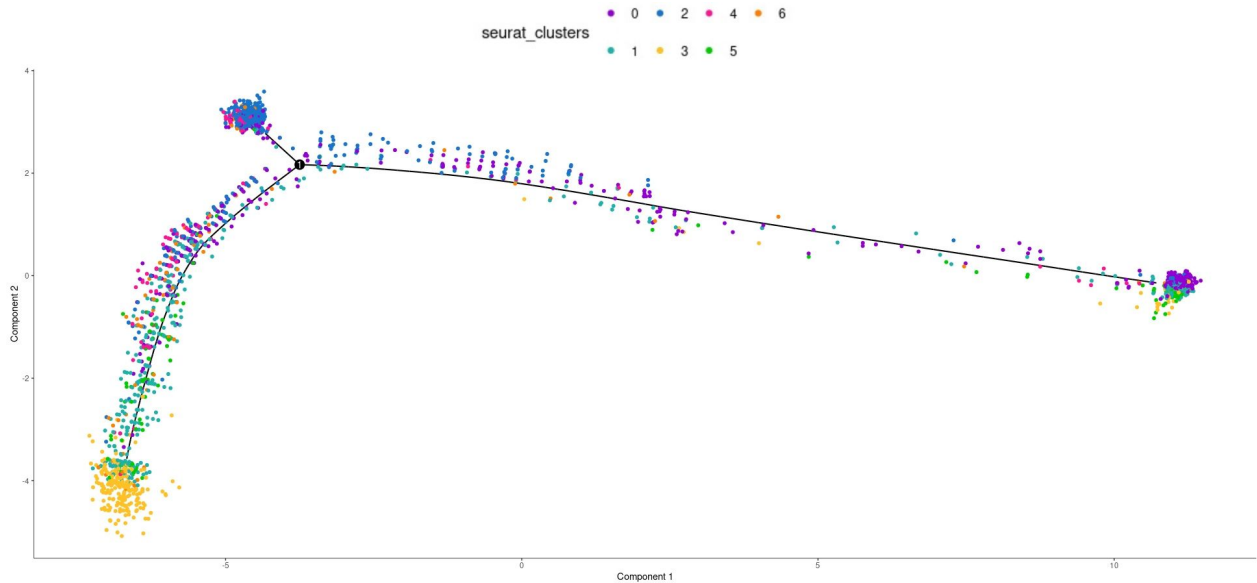


Figure E.24: Monocle2 Neutrophils trajectory colored by Seurat Clusters

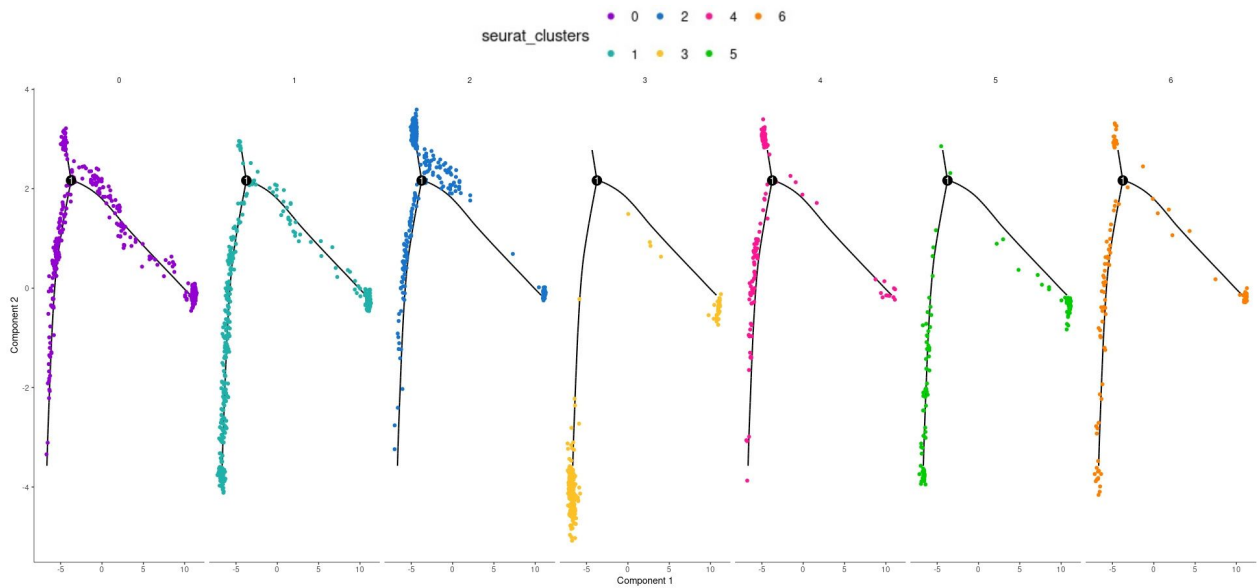


Figure E.25: Monocle2 Neutrophils trajectory divided by Seurat Clusters

Código E.20: R studio code extract – Neutrophils trajectory analysis using Monocle2 part 2

```

1
2 #- States -#
3 samplesIntegFltNeutrophils$State <- monocle_cds$State
4 DimPlot(samplesIntegFltNeutrophils, reduction = "umap", label = ↵
  ↵ FALSE, pt.size = 1.5, group.by = "State")
5
6 #- Pseudotime -#

```

```
7 samplesIntegFltNeutrophils$Pseudotime <- monocle_cds$Pseudotime
8 FeaturePlot(samplesIntegFltNeutrophils, features = 'Pseudotime', ←
  ↪ min.cutoff = 0, pt.size = 2, cols = c("royalblue", "brown2") ←
  ↪ )
```

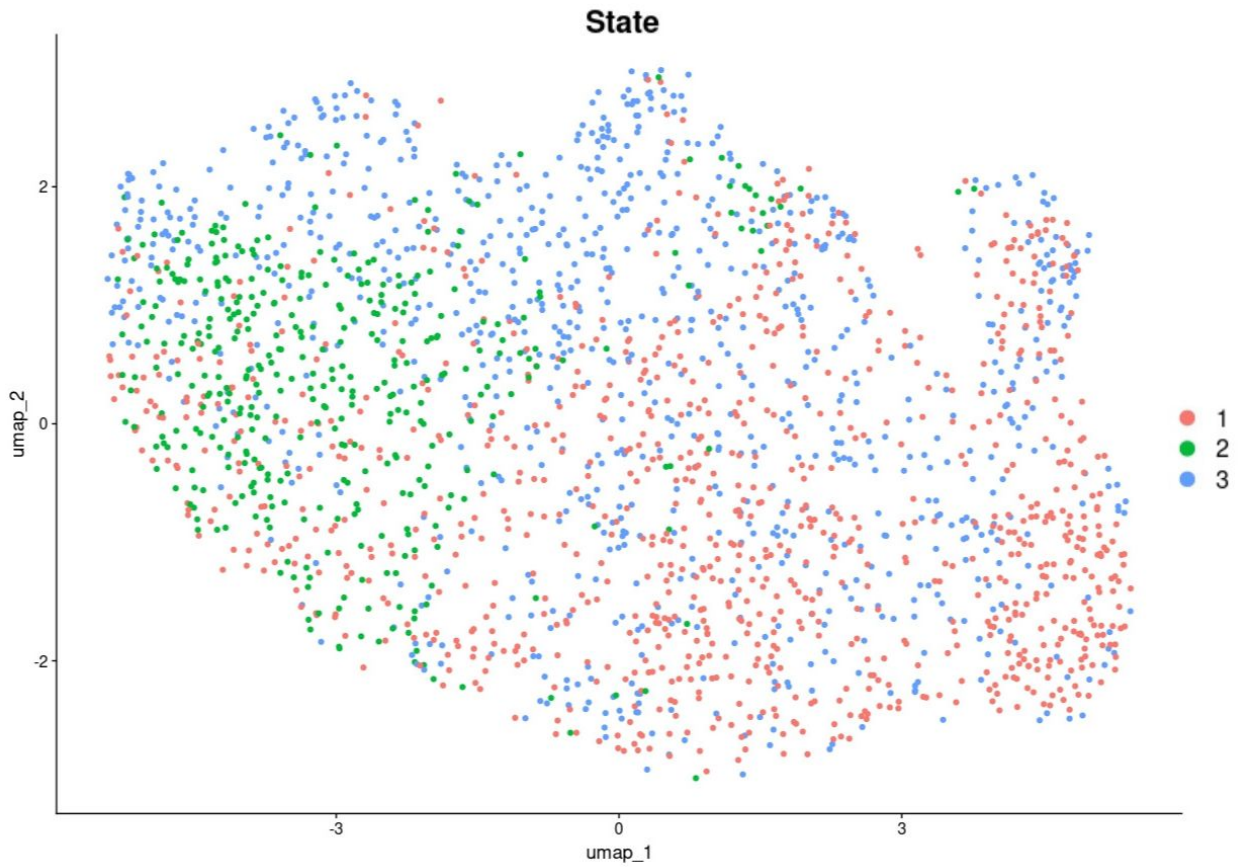


Figure E.26: Neutrophils UMAP divided by Monocle2 States

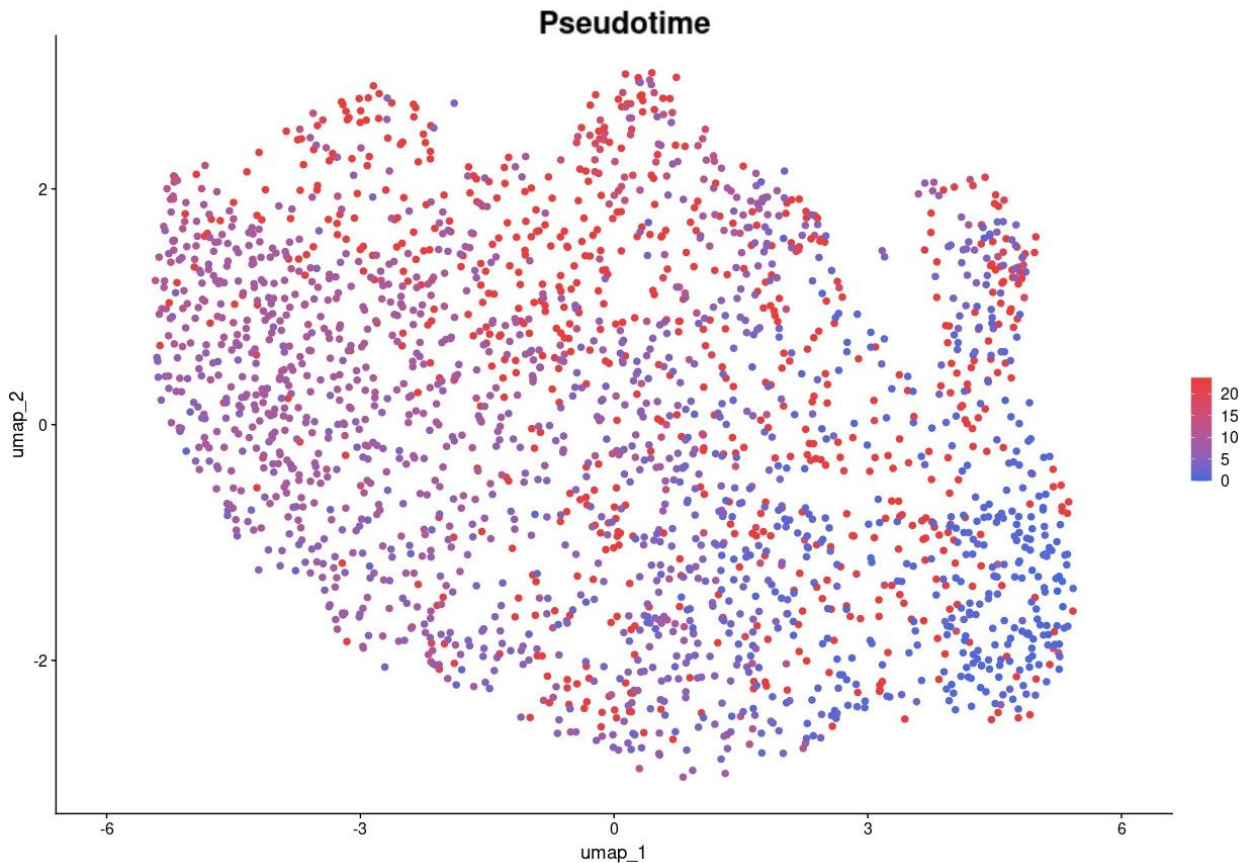


Figure E.27: Pseudotime in Neutrophils UMAP

Código E.21: R studio code extract – Alshetaiwi’s Mouse G-MDSC’s signature analysis

```

1 #- Alshetaiwi -#
2 DefaultAssay(object = samplesIntegFltGranulocytes) <- "SCT"
3 Datos_Granu_Alshetaiwi <- read.xlsx("GranuSignatures.xlsx", sheet = <-
  <- "Alshetaiwi")
4
5 Genes.MouseGMDSC <- Datos_Granu_Alshetaiwi[,1]
6 MouseGMDSC <- list(Genes.MouseGMDSC)
7
8 samplesIntegFltGranulocytes <- AddModuleScore(object = <-
  <- samplesIntegFltGranulocytes, features = MouseGMDSC, name = '
  <- MouseGMDSC')
9
10 #- Feature plot UMAP -#
11 FeaturePlot(samplesIntegFltGranulocytes, features = 'MouseGMDSC1', <-
  <- min.cutoff = 0, pt.size = 1.5, cols = c("royalblue", "
  <- brown2"))
12

```

```
13 #- Monocle trajectory -#  
14 monocle_cds$MouseGMDSC1 <- samplesIntegFltNeutrophils$MouseGMDSC1  
15  
16 plot_cell_trajectory(monocle_cds, color_by = "MouseGMDSC1")
```

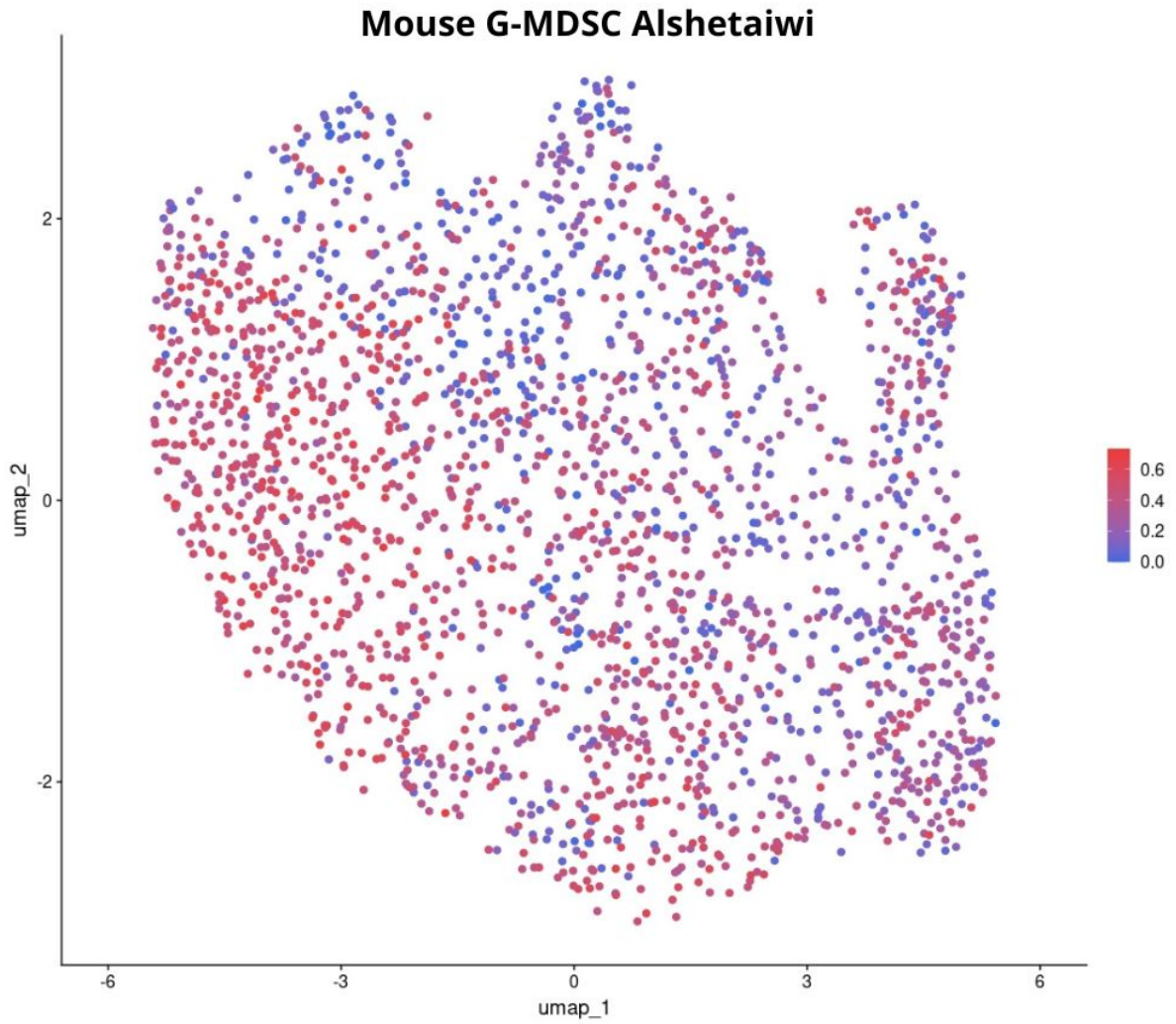


Figure E.28: Alshetaiwi's Mouse G-MDSC signature expression on UMAP

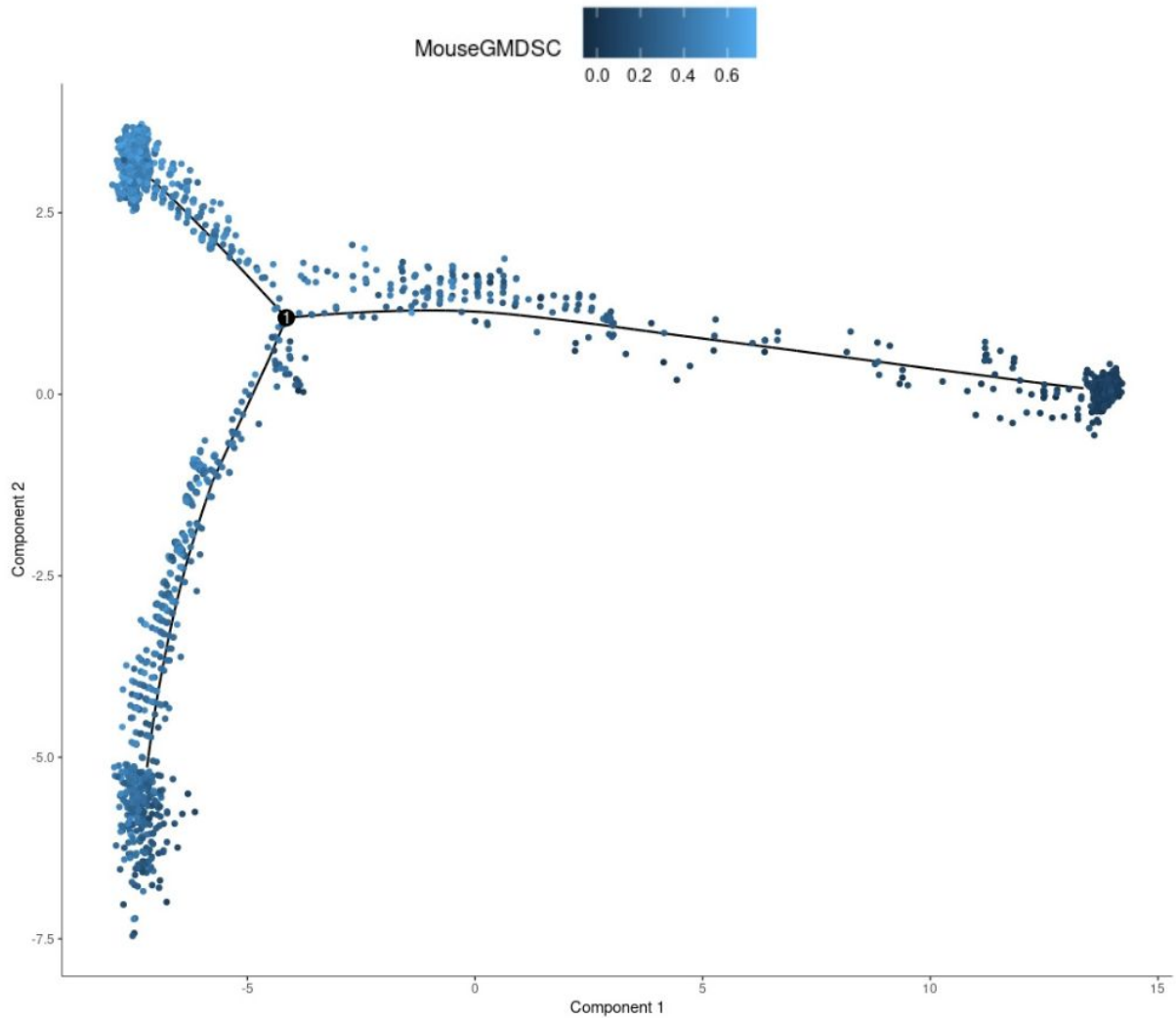


Figure E.29: Alshetaiwi's Mouse G-MDSC signature expression on Monocle trajectory

Código E.22: R studio code extract – Alshetaiwi's progenitor Neutrophil signature

```

1 #- Alshetaiwi Monocle -#
2 DefaultAssay(object = samplesIntegFltNeutrophils) <- "SCT"
3 Datos_Granu_Alshetaiwi <- read.xlsx("GranuSignatures.xlsx", sheet = <-
  <- "AlshetaiwiMonocle")
4
5 Genes.ProgenitorNeu <- Datos_Granu_Alshetaiwi[,4]
6
7 ProgenitorNeu <- list(Genes.ProgenitorNeu)
8
9
10 samplesIntegFltNeutrophils <- AddModuleScore(object = <-
  <- samplesIntegFltNeutrophils, features = ProgenitorNeu, name =<-

```

```

11     ↪ 'ProgenitorNeu')
12 #- Feature plot UMAP -#
13 FeaturePlot(samplesIntegFltNeutrophils, features = 'ProgenitorNeu1↪
    ↪ ', min.cutoff = 0, pt.size = 1.5, cols = c("royalblue", "↪
    ↪ brown2"))
14 #- Monocle trajectory -#
15 monocle_cds$ProgenitorNeu1 <- samplesIntegFltNeutrophils$↪
    ↪ ProgenitorNeu1

```

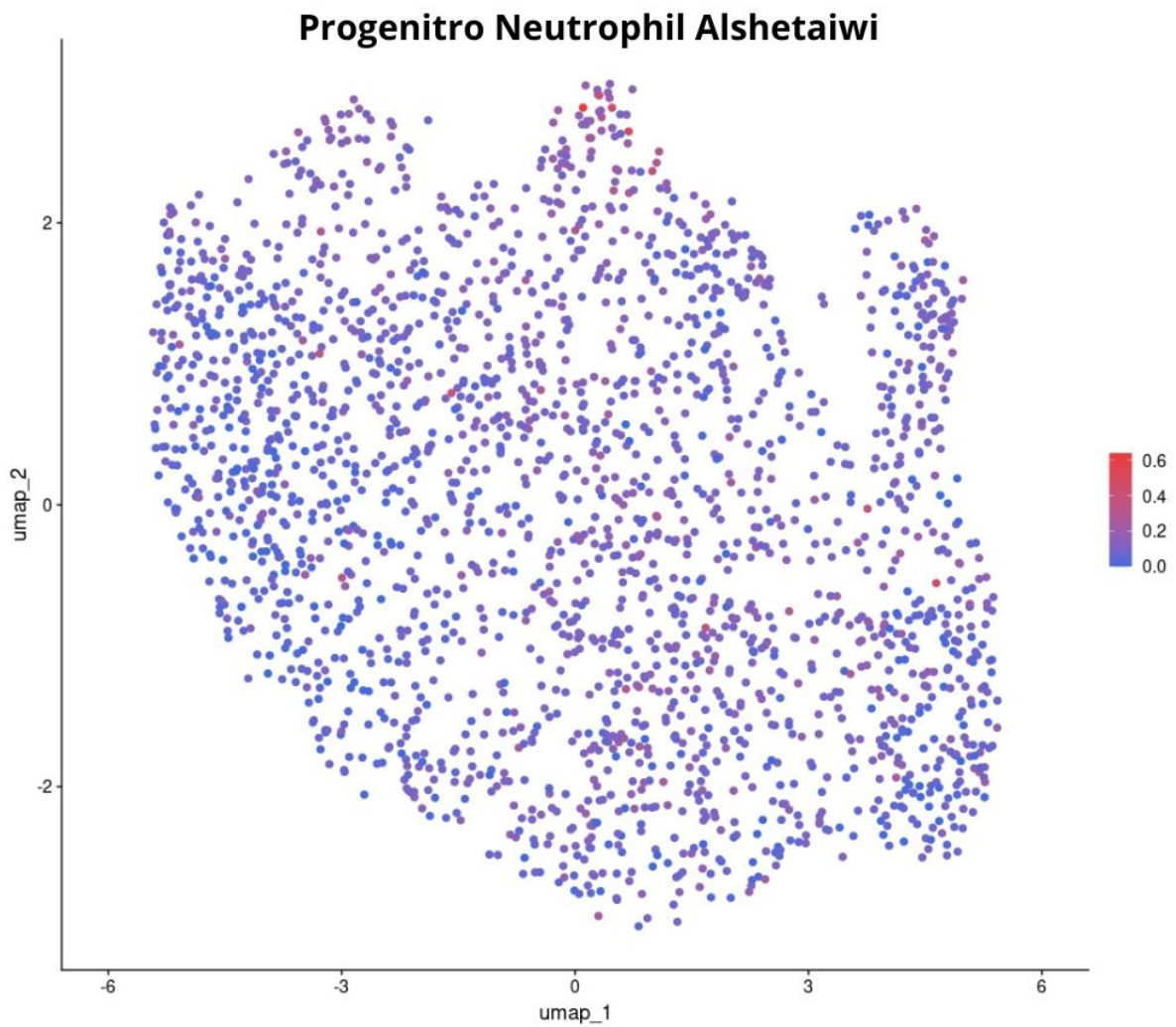


Figure E.30: Alshetaiwi's Progenitor Neutrophil signature expression on UMAP

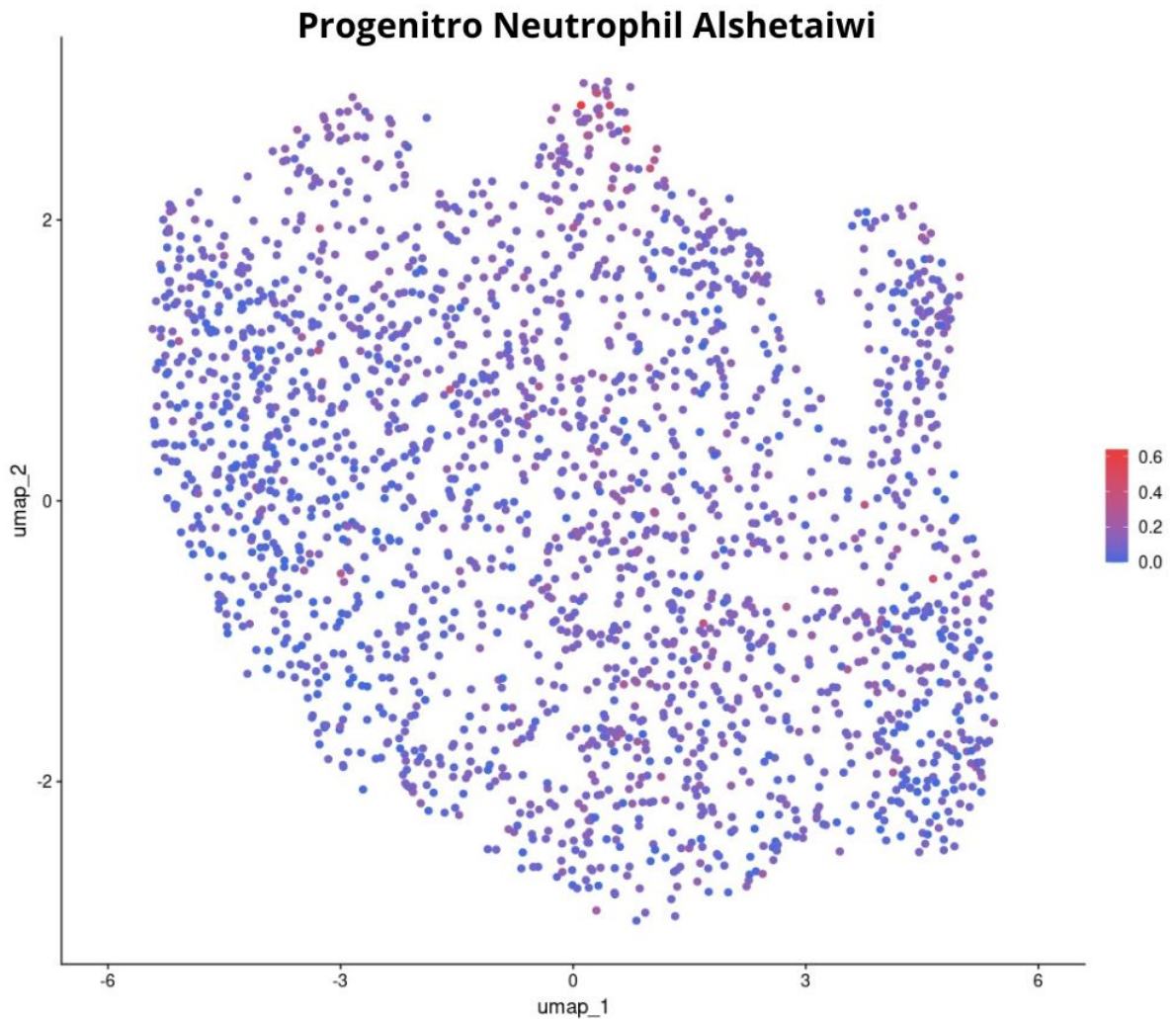


Figure E.31: Alshetaiwi's Progenitor Neutrophil signature expression on Monocle trajectory

Código E.23: R studio code extract – Neutrophils subpopulation division

```

1 #-- New classification --#
2 samplesIntegFltNeutrophils@active.ident <- ←
  ↪ samplesIntegFltNeutrophils$State
3
4 samplesIntegFltNeutrophils <- RenameIdents(object = ←
  ↪ samplesIntegFltNeutrophils, 1 = Neutrophils, 2 = G-MDSC, 3 = ←
  ↪ Mature Neutrophils)
5
6 DimPlot(samplesIntegFltNeutrophils, reduction = "umap", label = ←
  ↪ FALSE, pt.size = 2) + ggtitle('Neutrophils subpopulations')
```

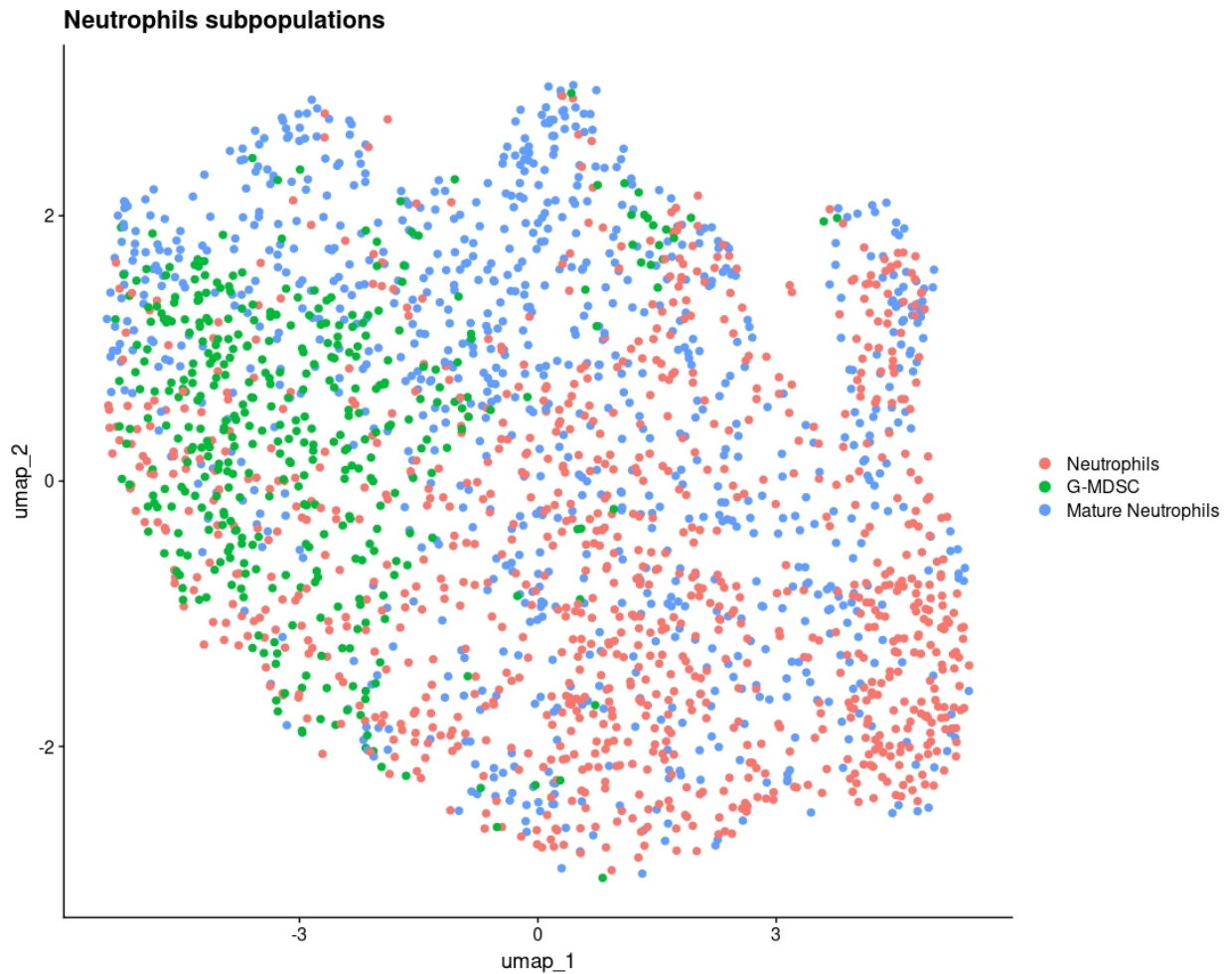


Figure E.32: Neutrophils subpopulations

Código E.24: R studio code extract – Neutrophils Barplots

```

1
2 samplesIntegFltNeutrophils[["Subpopulation"]] <- <->
  <-> samplesIntegFltNeutrophils@active.ident
3 dittoBarPlot(object = samplesIntegFltNeutrophils, scale = "percent<->
  <-> ", var = "Subpopulation", group.by = "Treatment", x.labels = <->
  <-> c("Untreated", "Treated"), xlab = "Condition", main = "")

```

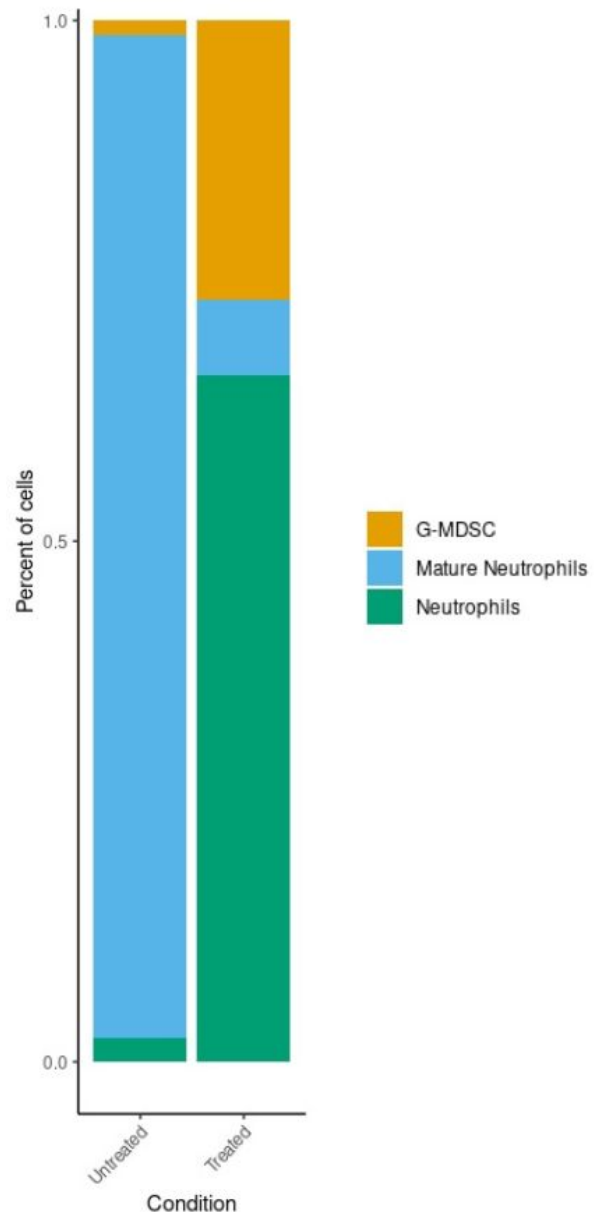


Figure E.33: Barplot of Neutrophils subpopulations percentage by Conditions

Annex F. Log2 Fold Change

To better quantify differences between conditions, the value of the log2 Fold Change for each population and subpopulation was calculated using the following equation:

$$\log_2FC = \frac{\log_2(Treated)}{\log_2(Untreated)} \quad (F.1)$$

F.1. General myeloid data

Table F.1: Number of cells in the different myeloid populations under untreated and treated conditions, alongside their log2FC value.

| | Untreated | Treated | Log2FC |
|-----------------------------|-----------|---------|--------|
| Monocyte/Macrophages | 9056 | 8769 | -0.05 |
| Dendritic cells | 973 | 613 | -0.67 |
| Granulocytes | 939 | 1568 | 0.74 |

After analyzing the Monocyte/Macrophage population and identifying the granulocytic contaminant cluster, these numbers change to:

Table F.2: Number of cells in the different myeloid populations under untreated and treated conditions, alongside their log2FC value.

| | Untreated | Treated | Log2FC |
|-----------------------------|-----------|---------|--------|
| Monocyte/Macrophages | 9056 | 8748 | -0.05 |
| Dendritic cells | 973 | 613 | -0.67 |
| Granulocytes | 939 | 1589 | 0.76 |

F.2. Dendritic cell data

Table F.3: Number of cells in the different dendritic cells subpopulations under untreated and treated conditions, alongside their log2FC value.

| | Untreated | Treated | Log2FC |
|-------------|-----------|---------|--------|
| cDC2 | 283 | 211 | -0.42 |
| DC3 | 59 | 136 | 1.20 |
| cDC1 | 160 | 26 | -2.62 |
| MoDC | 84 | 100 | 0.25 |
| pDC | 94 | 64 | -0.55 |

F.3. Monocyte and Macrophages data

Table F.4: Number of cells in the different monocyte and macrophage subpopulations under untreated and treated conditions, alongside their log2FC value.

| | Untreated | Treated | Log2FC |
|-----------------------------|-----------|---------|--------|
| Monocyte-Derived Mac | 1069 | 1227 | 0.20 |
| Rps8 Mac | 736 | 1156 | 0.65 |
| Ednrb Mono | 612 | 1246 | 1.03 |
| Fcrls Mac | 670 | 432 | -0.63 |
| Sell Mono | 302 | 732 | 1.28 |
| Cycling Mac | 746 | 217 | -1.78 |
| Cxcl9 Mac | 350 | 602 | 0.78 |
| Monocyte-Derived DC (Flt3-) | 467 | 407 | -0.20 |
| Pdgfc Mac | 400 | 336 | -0.25 |
| IFN Mac | 243 | 386 | 0.67 |
| Mreg | 249 | 333 | 0.42 |
| Resident Mono | 86 | 396 | 2.20 |

F.4. Granulocytes data

Table F.5: Number of cells in the different granulocyte subpopulations under untreated and treated conditions, alongside their log2FC value.

| | Untreated | Treated | Log2FC |
|------------------------------------|-----------|---------|--------|
| Neutrophils | 841 | 1414 | 0.75 |
| Eosinophils | 56 | 93 | 0.73 |
| Translationally active Neutrophils | 42 | 100 | 1.25 |

F.5. Neutrophils data

Table F.6: Number of cells in the different neutrophil subpopulations under untreated and treated conditions, alongside their log2FC value.

| | Untreated | Treated | Log2FC |
|--------------------|-----------|---------|--------|
| Neutrophils | 19 | 929 | 5.61 |
| G-MDSC | 11 | 378 | 5.10 |
| Mature Neutrophils | 805 | 103 | -2.97 |

Annex G. CD8+ T cell subsets analysis

Concerning T cell changes in the tumor microenvironment, it was observed that, under treatment conditions, there was an increase of CD8+ T cells *in vivo* (see Figure G.1).

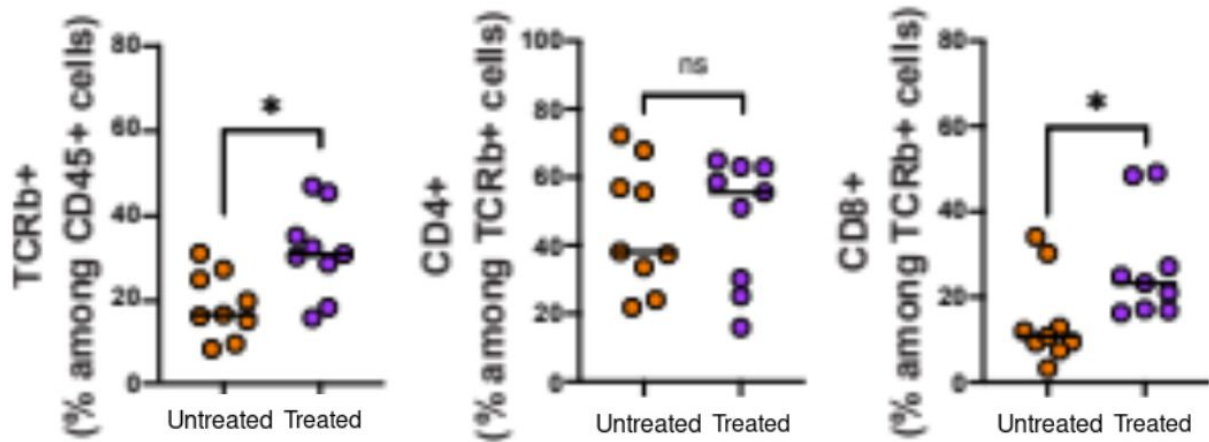


Figure G.1: Analysis of T cell population changes under untreated and treated conditions. Information provided by Institut Curie

Considering this, a study was done in collaboration with Institut Curie, where CD8+ datasets under treated and untreated conditions were analysed. Similar to the work done in the present work, R studio was used to analyse this data and determine the changes in subpopulations. This analysis determined that a total of 12 subpopulations were present, which can be grouped into three main categories; Naive and central memory (CM), effectors and exhausted. Under treatment condition, an increase in naive/CM cells was observed, while exhausted cells decreased. More importantly, a particular subset, known as progenitor exhausted CD8+ was observed to increase in treatment condition (see Figure G.2). This type of cell gives rise to exhausted T cells and is the main target of treatments that prevent cell exhaustion.

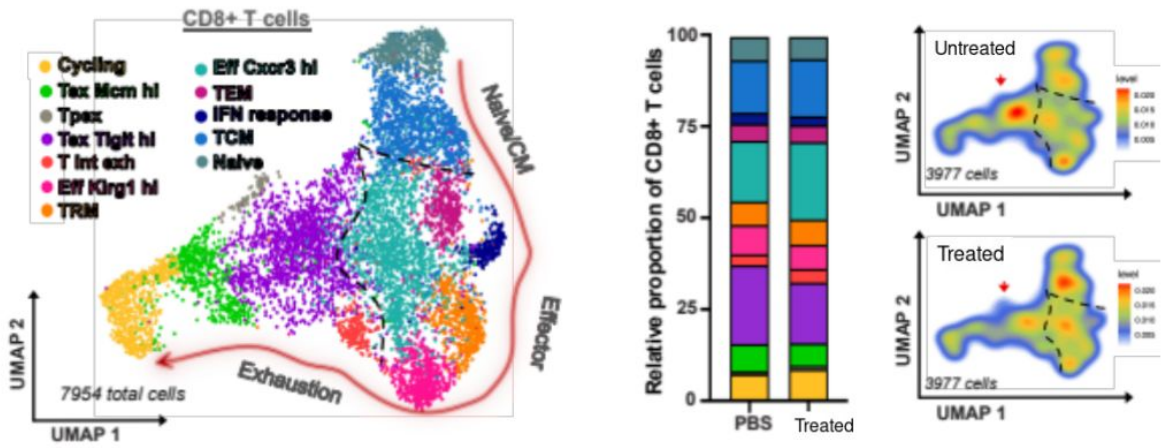


Figure G.2: CD8+ subpopulations change analysis under untreated and treated conditions

This observation was further confirmed by studies done by Institut Curie, in which it was observed that *in vivo* Tpex (progenitor exhausted T cells) increased under treated conditions (see Figure G.3).

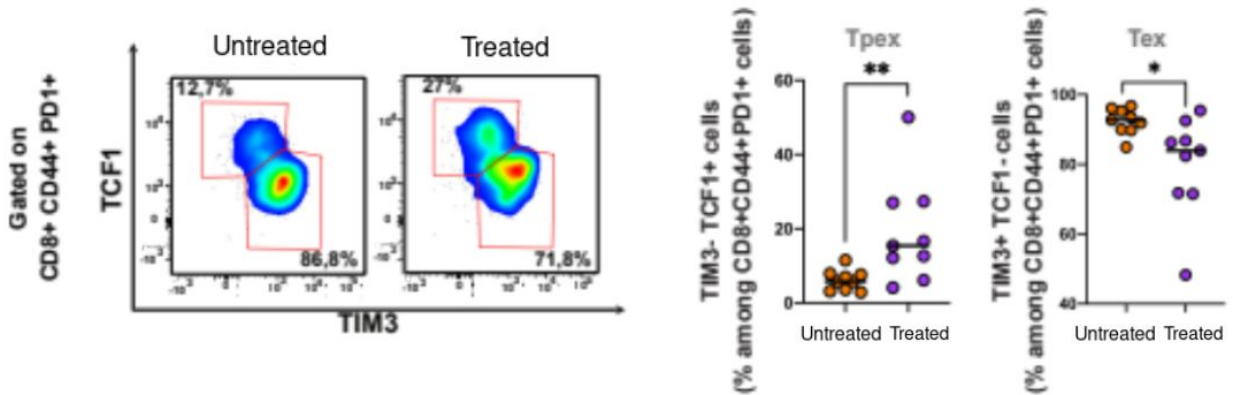


Figure G.3: Progenitor exhausted (Tpex) and terminally exhausted (Tex) CD8+ T cells change analysis under untreated and treated conditions. Information provided by Institute Curie