1 Introduction

One of the central cellular processes underlying development is transcriptional regulation. Modelling the dynamics of gene regulation is therefore essential to better understand why a cellular dynamic processes progresses through several steps, and what goes wrong in the case of disease. Regulatory dynamics is classically studied using time series data [1]. When dynamic processes progress asynchronously, such as in hematopoiesis, time series data are usually obtained by sorting different transition states and assessing bulk gene expression and transcription factor binding within the population [2, 3, 4, 5]. Alternatively, time series data can also be generated by synchronising the dynamic process between cells. However, issues with time-resolution, heterogeneity and good *in vivo* synchronisation models can often limit the predictive power of the dynamic models of gene regulation which can be constructed [1].

Network inference (NI) methods are computational tools which use large omics datasets to predict which genes are regulated by which transcription factors. While accuracy of the network of predicted regulatory interactions is lower in comparison to experimental validation techniques, NI methods offer an unbiased and high-throughput insight into the regulatory dynamics of a biological system. The output of a network inference is thus a graph, where nodes represent genes and edges denote a regulatory interaction between a regulator and a target gene. Interactions have two properties: its regulatory strength (a positive real value) and its effect (promoting or repressing).

Several studies have highlighted how some regulatory interactions can be very dynamic while others show evidence of being static during consecutive developmental stages [6, 7]. Since regulatory interactions are context-dependent [8], attempting to create an accurate model of those processes by inferring a static regulatory network may have limited relevance. Case-wise NI methods[1] avoid predicting a static GRN and instead infer one GRN per cell (or per sample, for bulk omics data).

In order to compute a case-wise GRN for a single sample, Kuijjer et al. [9] and Liu et al. [10] employ similar strategies, namely by computing the difference of computing a static GRN for all the cases, and computing a static GRN for all the cases minus one. Since this procedure needs to be repeated for every case in the dataset, and because NI methods are already amongst the most computationally intensive analyses to perform on omics data, this methodology is not applicable for large omics datasets. Another case-wise NI method, SCENIC [11] infers case-wise GRNs by first inferring a static GRN using GENIE3 [12]. GENIE3 is a static NI method which uses Random Forests (RFs) [13] variable importance scores to prioritise candidate regulators for a particular target gene. SCENIC then post-processes the static GRN to determine whether an interaction is enriched for particular cases, resulting in a case-wise GRN.

Thus, while several case-wise NI methods have already been proposed, their implementation consisted of post-processing a static GRN to arrive at a case-wise GRN. As such, these

[1] Case-wise NI is sometimes also called sample-specific NI or case-specific NI.

methods will most likely recover the interactions that are prevalent in the whole population, and will miss interactions that are specific to only a sub-population.

In this work, we introduce `bred`, the first 'true' case-wise NI method. It uses a modified version of the RF variable importance scores used by GENIE3 and SCENIC to compute importance values for each profile and each interaction separately, as well as predict the effect of each interaction (activating or repressing). We generate case-wise GRNs – or case-wise regulomes – for 14'963 profiles from The Cancer Genome Atlas (TCGA) [14], resulting in 6'464'915 predicted case-wise interactions. We analyse these case-wise regulomes by clustering them and detecting highly activated interactions.

2 Results

At the time of writing, the TCGA database contains 14'963 profiles from 44 different cancer entities. We used the `bred` algorithm to infer case-wise regulomes for each of these profiles. In total, we detected 73'140 unique interactions and, on average, 7'231 active interactions per profile. Dimensionality reduction and clustering of the case-wise regulome data provides a visual overview of the different subpopulations (Figure 1).

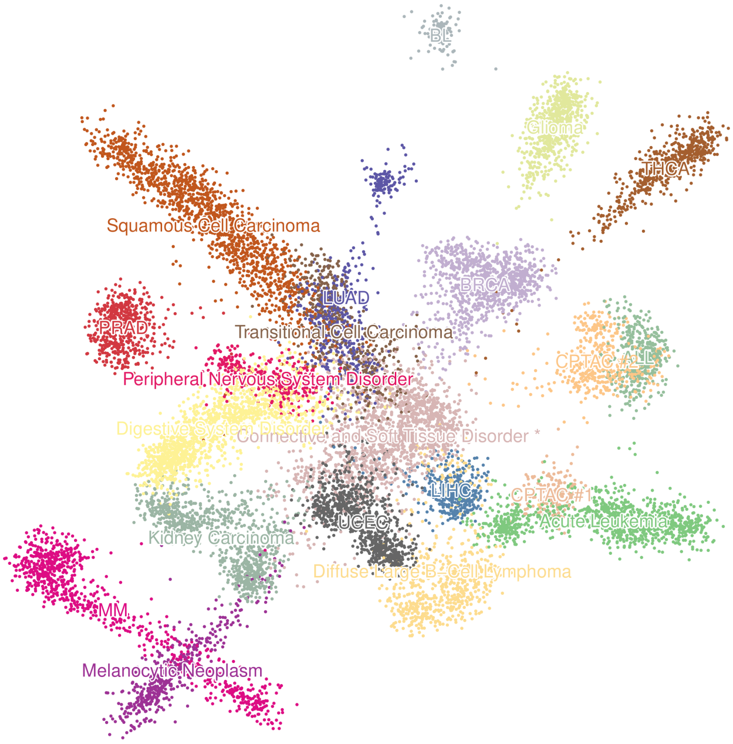


Figure 1: Visualisation of 14'963 case-wise regulomes of cancer profiles from The Cancer Genome Atlas project. Dimensionality reduction of the case-wise regulomes was performed by applying Fruchterman-Reingold [15] on the k-nearest-neighbour graph ($k = 100$) of highly similar regulomes (Spearman correlation). The samples were clustered with Louvain clustering [16] and each cluster was assigned an ontology term from the NCI thesaurus ontology [17] which best fits the sample in the cluster. Clusters for which the term has a positive predictive value (PPV) lower than 0.5 are marked with an asterisk (*). More detailed information on the enrichment of particular terms or metadata for each of the clusters can be found in Figures 6–12.

For each cluster, we computed the average importance of each interaction. Retaining the 100 strongest interactions (Figure 2) shows that different cancer entities have vastly different regulomes, yet often have shared regulators and interactions (Table 1).

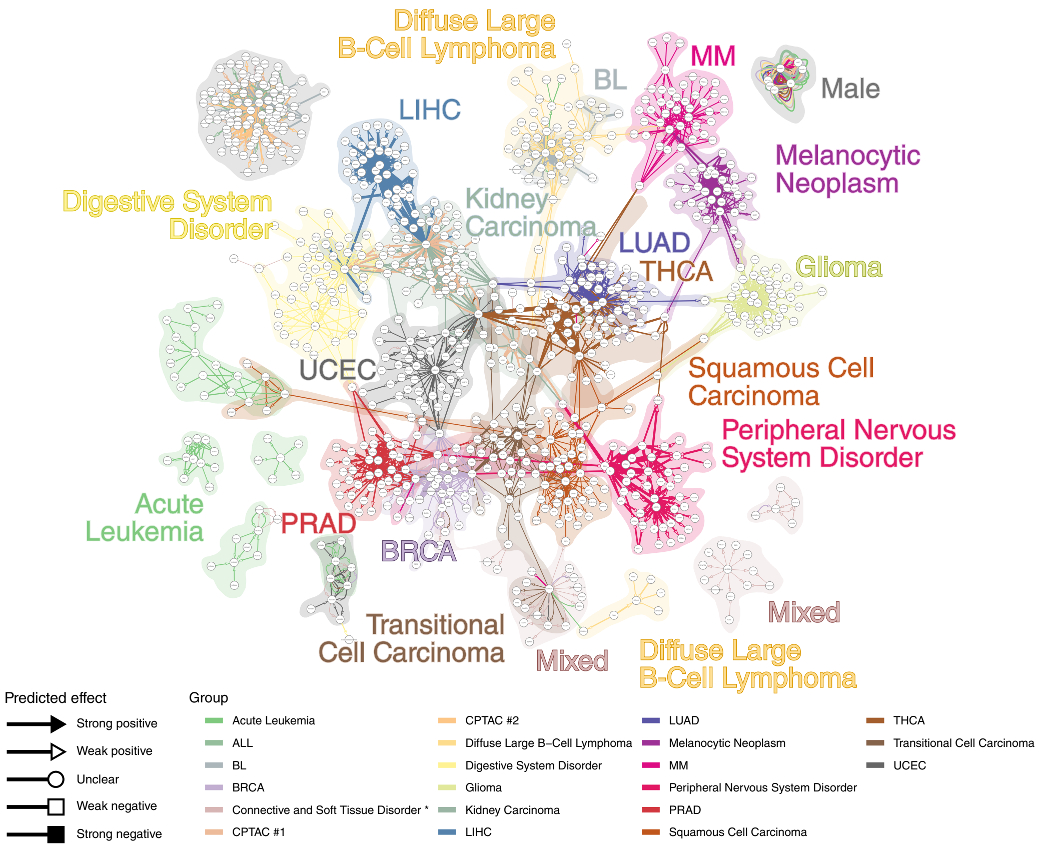


Figure 2: Visualisation of the strongest interactions per cluster shows both pathways distinct to particular cancer entities as well as pathways common to multiple cancer entities.

Performing a more in-depth analysis of particular clusters can reveal distinct sub-populations within. For example, when comparing samples of normal and healthy tissue within breast carcinoma (Figure 3), a significant number if interactions are predicted to have been turned off (top), while others have been turned on (middle, bottom). Similar observations can be made for other clusters, such as for melanocytic neoplasm (Figure 13) and kidney carcinoma (Figure 14).

3 Discussion

The `bred` algorithm is a novel approach for directly computing case-wise regulomes for single-cell omics and bulk omics profiles alike. We used `bred` to infer case-wise regulomes for 14'963 cancer profiles from The Cancer Genome Atlas project. Analysing the results using common omics algorithms (dimensionality reduction, clustering, enrichment) has shown that regulatory interactions are often specific to particular cancer types, but can also be shared between several cancer types. Ultimately, this analysis has resulted in a list of candidate oncogenic genes and interactions per cancer type, though further analysis and valida-

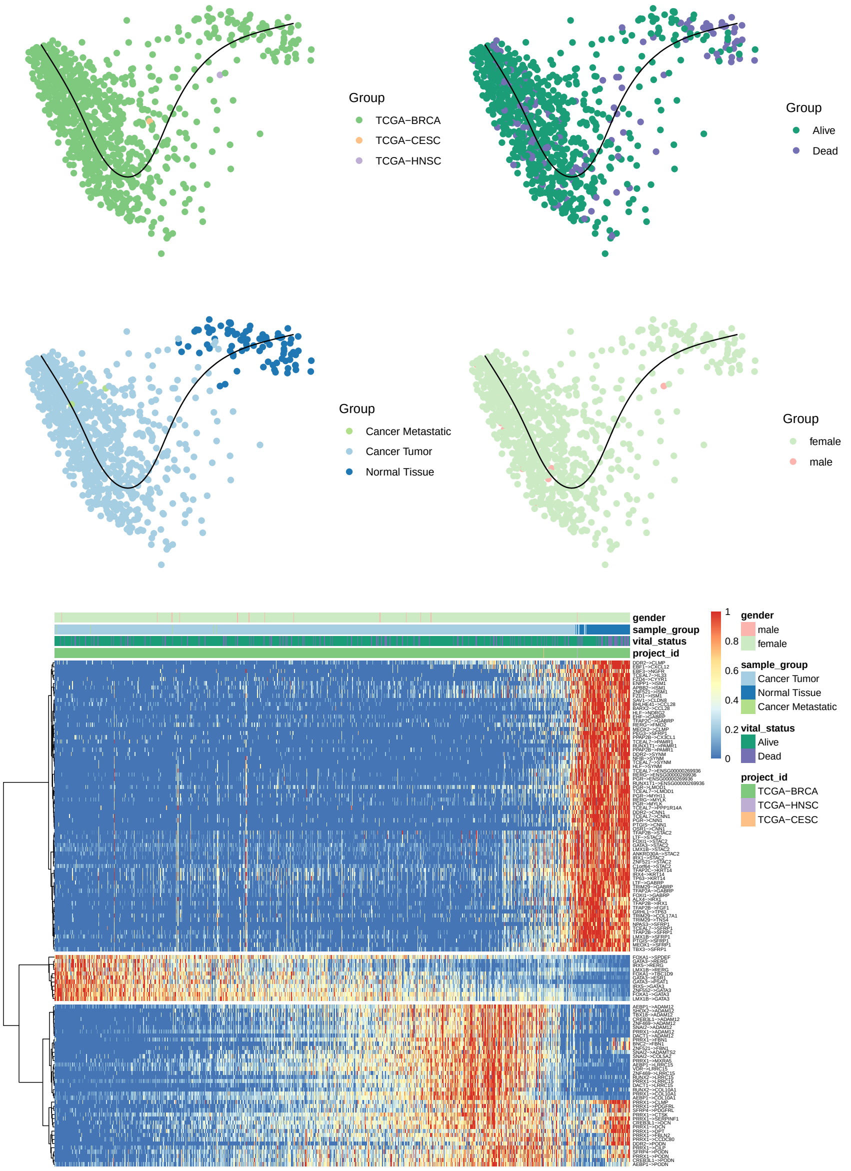


Figure 3: In-depth view of cluster 17, breast carcinoma. Top: A dimensionality reduction of the samples, coloured according to multiple sources of meta-data. **Bottom:** The samples were ordered linearly with SCORPIUS (see trajectory in **Top**) in order to visualise regulome activity in the form of a heatmap.

tion is required to confirm their oncogenicity.

Future perspectives include applying it on single-cell omics data (e.g. from Tabula Muris [18]) and benchmarking the algorithm against *in silico* datasets (e.g. produced by `dyngen` [19]).

4 Methods

4.1 Inferring case-wise regulomes

The task of inferring a static GRN (Figure 4A) can be reduced to a simpler problem, namely: for every target T, predict which of the potential regulators R_i regulate T (Figure 4B). This simplification allowed GENIE3 [12] to use Random Forest's [13] feature importance scores for inferring GRNs. Namely, a Random Forest is trained to predict the expression of a target gene of interest from the expression of potential regulators. The resulting Random Forest inherently allows to extract a feature importance score by observing the effect of each regulator in making a good prediction for the target expression. As in GENIE3, the target expression is first scaled to normalise feature importance scores across different targets.

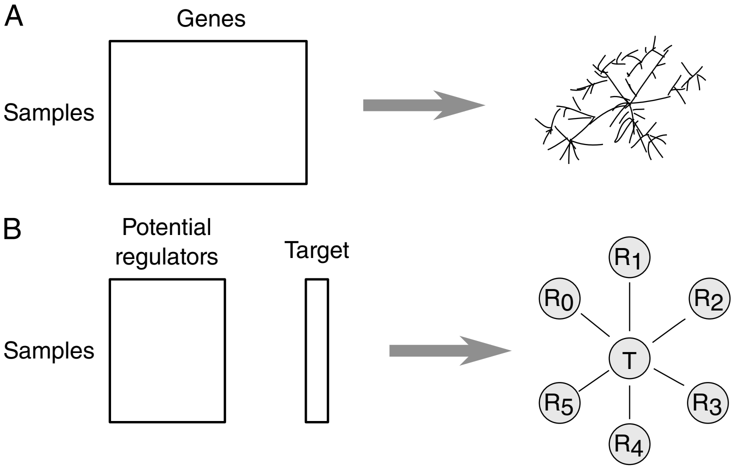


Figure 4: A: Inferring a gene regulatory network from an omics dataset can be reduced to a simpler problem. **B:** Given the expression of a target of interest and a set of potential regulators, predict which regulators regulate the target gene.

We make the same simplification in order to build case-wise GRNs, also using Random Forests to compute the feature importance scores. A Random Forest consists of K trees, each of which produces feature importance scores, and the feature importance scores of a forest is simply the mean feature importance scores of each of the trees.

Computing the case-wise feature importances of a tree consists of the following 8 steps (Figure 5). The 'randomness' of a Random Forest is due to only using a subset of the samples in the dataset in order to build a single decision tree. The samples are split into two groups, the 'in-bag' data and the 'out-of-bag' data (Figure 5A). A decision tree [20] is trained on the in-bag expression of the potential regulators in trying to predict the in-bag target gene expression (Figure 5B). The target expression of the out-of-bag samples is predicted using the decision tree (Figure 5C), and the squared error between the real and target expression is computed (Figure 5D). For each sample in the out-of-bag set, this vector represents how well the decision tree was able to predict the expression of the target gene.

The next few steps are repeated for every potential regulator R_i. Within the out-of-bag samples, the expression of R_i is randomly shuffled. The target expression of the out-of-bag samples is again calculated (Figure 5F), as well as the squared error between the real target expression and the predicted expression is calculated (Figure 5G). The importance of regulator R_i for an out-of-bag sample S_j is defined as the increase in squared error between the predicted target expression and the real target expression, after perturbing the expression of R_i (Figure 5H).

Steps F-G are repeated for every potential regulator R_i. By aggregating all of the feature importance scores over all the samples, regulators and targets, we obtain an M-by-N-by-P tensor[2].

A moderately-sized dataset could contain $M = 10'000$ samples, $N = 2'000$ regulators, and $P = 10'000$ target genes. Due to memory constrains, only interactions with an average importance value (across all samples) higher than a minimum threshold are retained.

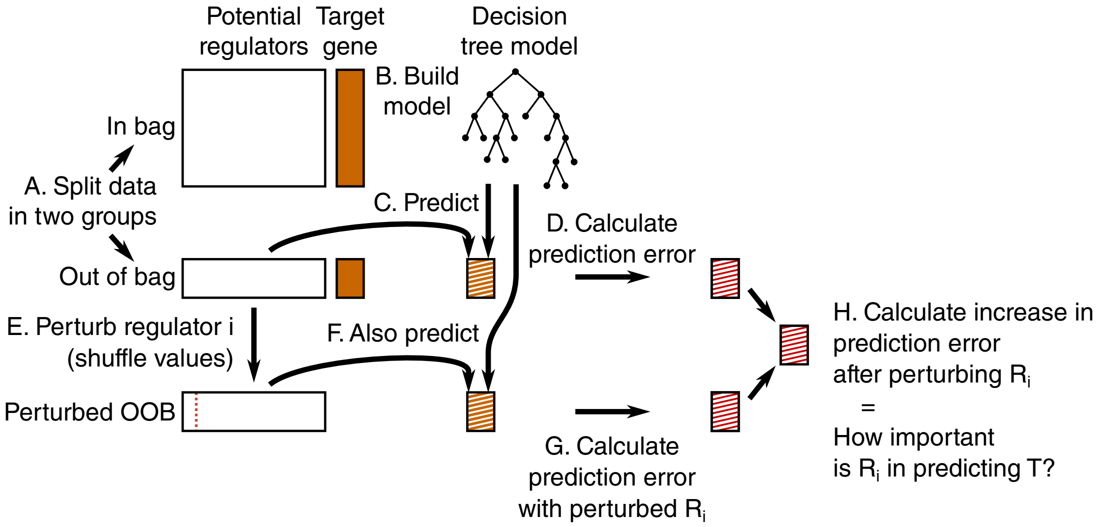

Potential regulators
Target gene
Decision tree model
B. Build model
In bag
A. Split data in two groups
Out of bag
C. Predict
D. Calculate prediction error
E. Perturb regulator i (shuffle values)
F. Also predict
Perturbed OOB
G. Calculate prediction error with perturbed R$_i$
H. Calculate increase in prediction error after perturbing R$_i$
=
How important is R$_i$ in predicting T?

Figure 5: Calculating the feature importance score for one decision tree and one target consists of 8 distinct steps. **A:** Randomly split the data into two groups, the in-bag data and the out-of-bag data. **B:** The in-bag data is used to train a decision tree to try to predict the expression of the target gene from the expression values of the regulators. **C:** The decision tree is used to predict the gene expression of the target gene of the out-of-bag samples. **D:** Sample-specific squared error values are computed. **E:** Repeat steps E-H for every regulator R_i. Perturb the expression of regulator R_i in the out-of-bag samples. **F:** Again predict the gene expression of the target gene with the perturbed expression values. **G:** Again compute the sample-specific squared error values. **H:** The difference between the prediction error on the perturbed dataset versus the prediction error on the unperturbed is the importance in R_i in predicting T

To compute the case-wise GRNs, we implemented the abovementioned methodology in C++ in a modified version of the `ranger` R/C++ package [21].

4.2 Predicting the effect of an interaction

To predict the effect of a potential regulator R_i on a target gene T for a given tree, the Pearson correlation is calculated between the difference in regulator expression (before and after shuffling the values), and the difference in target expression prediction.

$$\begin{aligned} \text{effect}(R_i \rightarrow T) &= \text{cor}(x, y), \\ \text{with } x &= \text{expr\_shuffled}[:, R_i] - \text{expr}[:, R_i], \\ \text{and } y &= \text{predict}(\text{tree}, \text{expr\_shuffled}) - \text{predict}(\text{tree}, \text{expr}). \end{aligned}$$

The Pearson correlation between two variables x and y is usually defined as shown in Equation 1. Computing r_{xy} for each (regulator, target) pairs, across all trees, would require

[2] This is the origin of the name of the method, "`bred`".

storing large amounts of data.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) \times (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

However, by rearranging the formula, it can be defined as Equation 2.

$$r_{xy} = \frac{\sum(x_i \times y_i) - \sum x \times \sum y/n}{\sqrt{(\sum x_i^2 - (\sum x)^2/n)} \times \sqrt{(\sum y_i^2 - (\sum y)^2/n)}} \tag{2}$$

For every regulator R_i during a perturbation in a given tree, only 6 values need to be stored, namely $A = \sum x_i$, $B = \sum y_i$, $C = n$, $D = \sum x_i \times y_i$, $E = \sum x_i \times x_i$, and $F = \sum y_i \times y_i$.

For every (regulator, target) pair, these values are summed, and the r_{xy} is calculated as shown in Equation 3.

$$r_{xy} = \frac{D - A \times B/C}{\sqrt{(E - A^2/C)} \times \sqrt{(F - B^2/C)}} \tag{3}$$

The following cutoffs were used to determine the effect.

- Strong negative: $r_{xy} < -0.4$
- Weak negative: $-0.4 \leq r_{xy} < -0.2$
- Unclear: $-0.2 \leq r_{xy} \leq 0.2$
- Weak positive: $0.2 < r_{xy} \leq 0.4$
- Strong positive: $0.4 < r_{xy}$

4.3 Clustering of case-wise GRNs

To perform downstream analysis on the cases, first a k-nearest neighbour (kNN) graph of the cases is computed. In order for the kNN graph to better emphasise similarities in GRNs rather than absolute euclidean distances, we first reduce the dimensionality of the case-by-interaction matrix to case-by-20 matrix using Landmark Multi-Dimensional Scaling [22] with a Spearman rank distance metric.

Next, KD-trees are used to calculate the kNN graph efficiently. The cases in the dataset are visualised and clusted using the Fruchterman-Reingold [15] and Louvain clustering [16], respectively.

The following R packages provided implementations for each of these algorithms: lmds, RANN, igraph [23].

4.4 Visualising clustered GRNs

After Louvain clustering, the interactions of the 50 interactions with highest mean importance per cluster are retained. These interactions are visualised in Cytoscape [24], in which nodes depict genes, edges depict predicted regulatory interactions, coloured according to which cluster they are predicted for. The shape of the arrow denotes the predicted effect of the regulatory interaction.

5 Supplementary information

Table 1: Prioritisation of genes, ranked according to page-rank value. The majority of these genes are already known to be involved in oncogenesis (data not shown).

Cluster name	Top 4 genes, page rank
Acute Leukemia	HSPA1A, HSPA1B, HOXA10, HOXA9
ALL	DDX3Y, XIST, RN7SL128P, TXLNGY
BL	KIAA0226L, XIST, TCL1A, NUGGC
BRCA	LINC00993, IRX5, ATP8A2P1, ENSG00000234918
Connective and Soft Tissue Disorder *	HOXA11, HOXC9, HOXA10, HOXC10
CPTAC #1	SLC22A2, CUBN, XIST, UMOD
CPTAC #2	RN7SL752P, SCARNA13, XIST, RNY1
Diffuse Large B-Cell Lymphoma	PLA2G2D, ENSG00000224137, CCL25, MS4A1
Digestive System Disorder	MEP1A, LGALS4, MUC13, CDX1
Glioma	NCAN, GPM6A, MBP, OLIG2
Kidney Carcinoma	ACSM2A, SLC22A2, NAT8, UMOD
LIHC	AGXT, ITIH3, ITIH1, HRG
LUAD	SFTPA2, SFTPD, NAPSA, SFTA2
Melanocytic Neoplasm	MLANA, RPS4Y1, PMEL, TYR
MM	CD96, GPRC5D, FGFR3, IGLV1-41
Peripheral Nervous System Disorder	TH, DBH, PENK, HAND2
PRAD	TRGC1, NKX3-1, ACPP, RPL7P16
Squamous Cell Carcinoma	TP63, SPRR2F, RPS4Y1, CDKN2B
THCA	ENSG00000240237, TG, TPO, RPS4Y1
Transitional Cell Carcinoma	DHRS2, UPK2, PADI3, VGLL1
UCEC	DLX5, MSX1, HOXB5, HOXB8

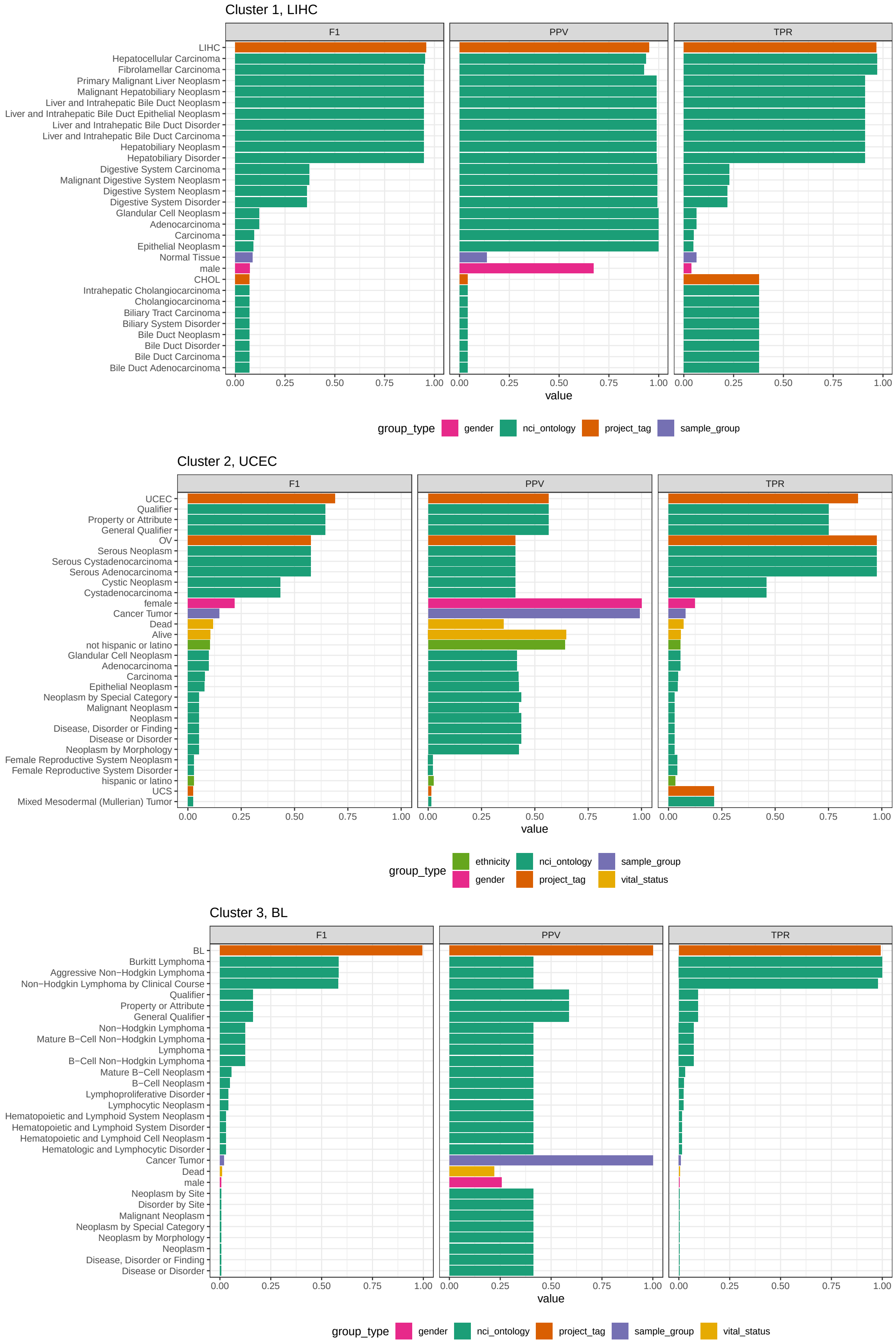


Figure 6: Prioritisation of NCI terms and other forms of meta-data used to annotate clusters 1–3. Cluster 2 is mainly made up of disorders in female reproductive organs, and should have been labelled as such.

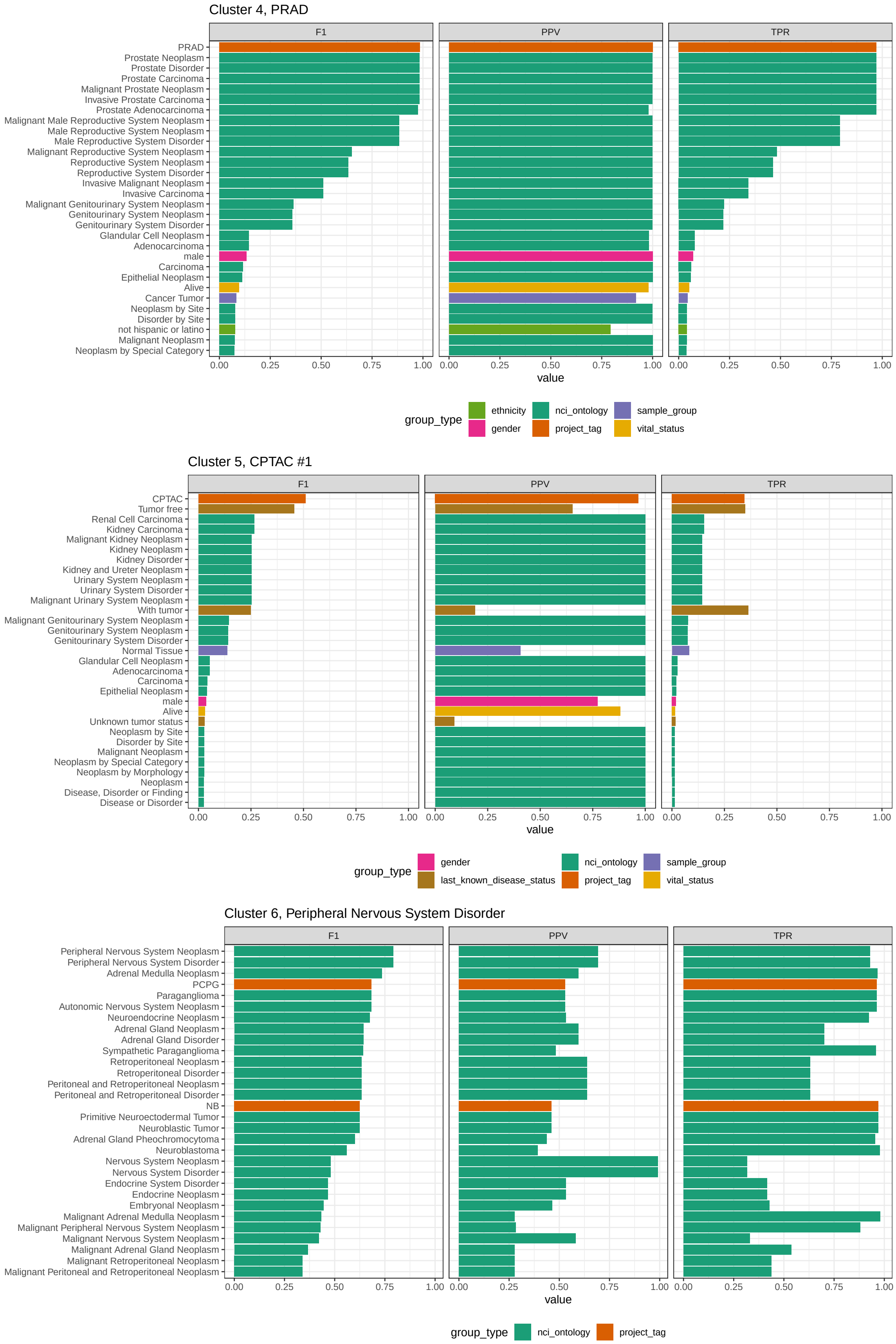


Figure 7: Prioritisation of NCI terms and other forms of meta-data used to annotate clusters 4–6. Cluster 5 is mainly made up of renal cell carcinoma, and should have been labelled as such. Cluster 6 is aptly named, but mainly consists of neuroblastoma (NB), pheochromocytoma (PC), and paraganglioma (PG) disorders.

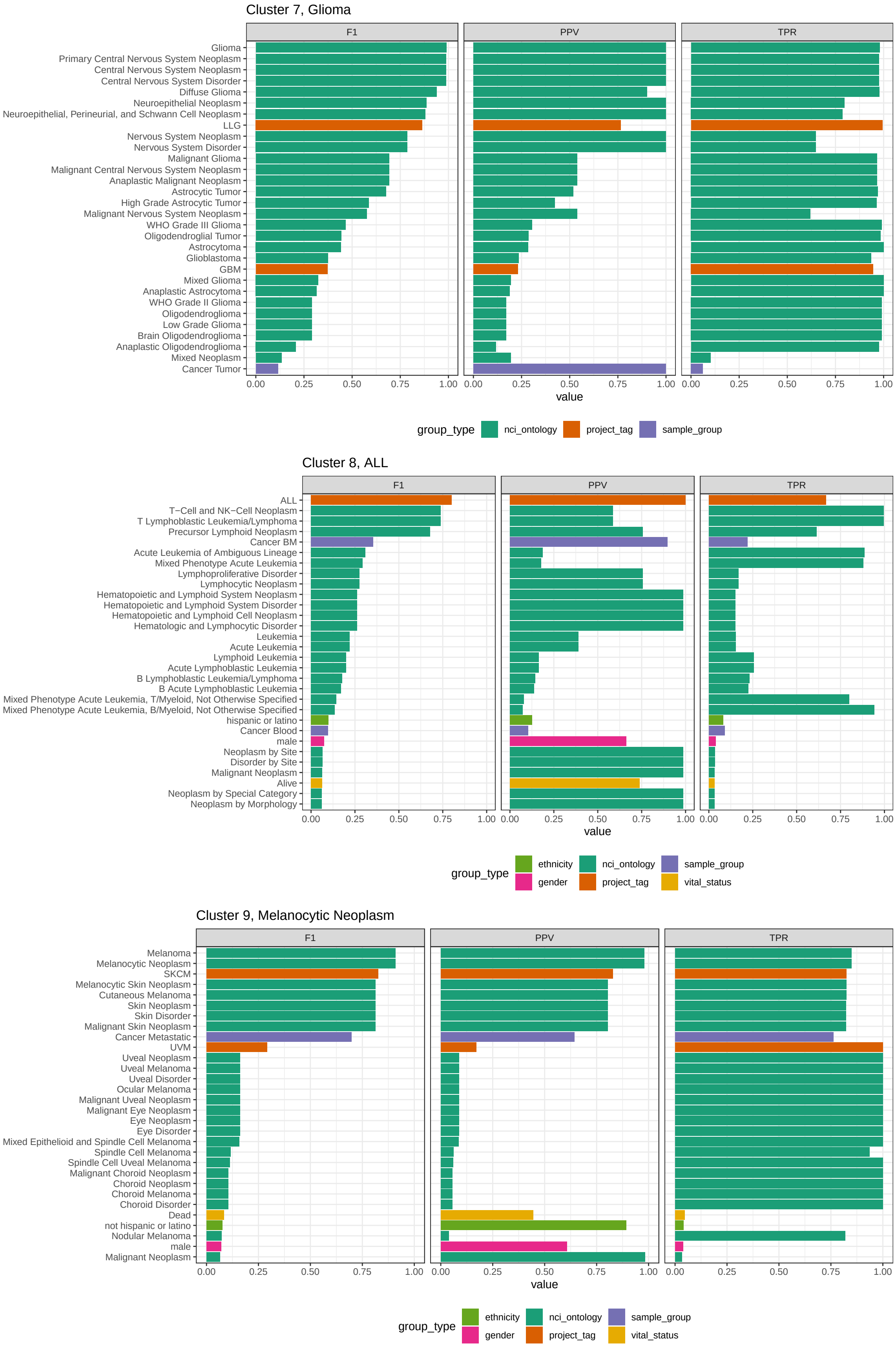


Figure 8: Prioritisation of NCI terms and other forms of meta-data used to annotate clusters 7–9.

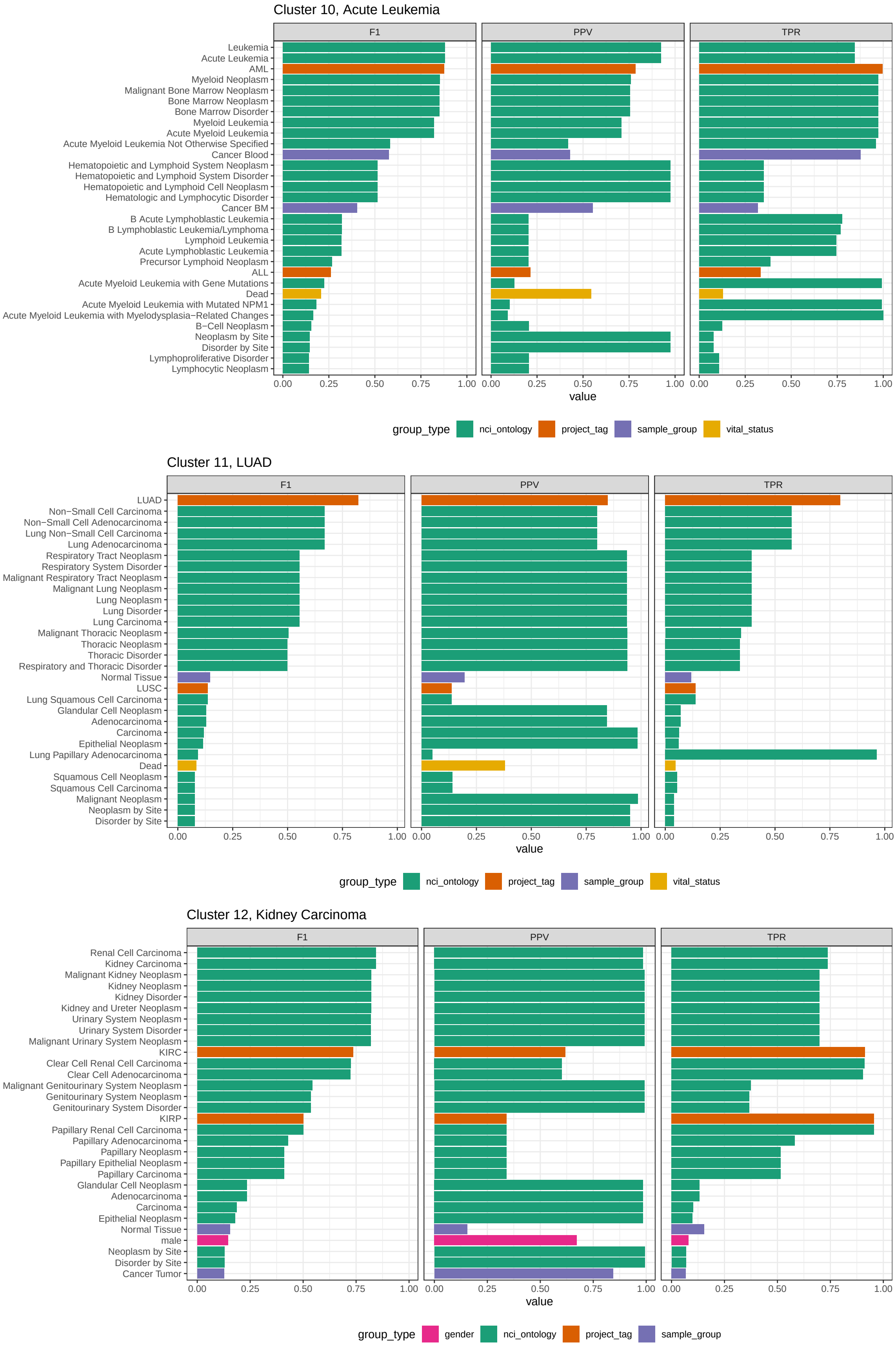
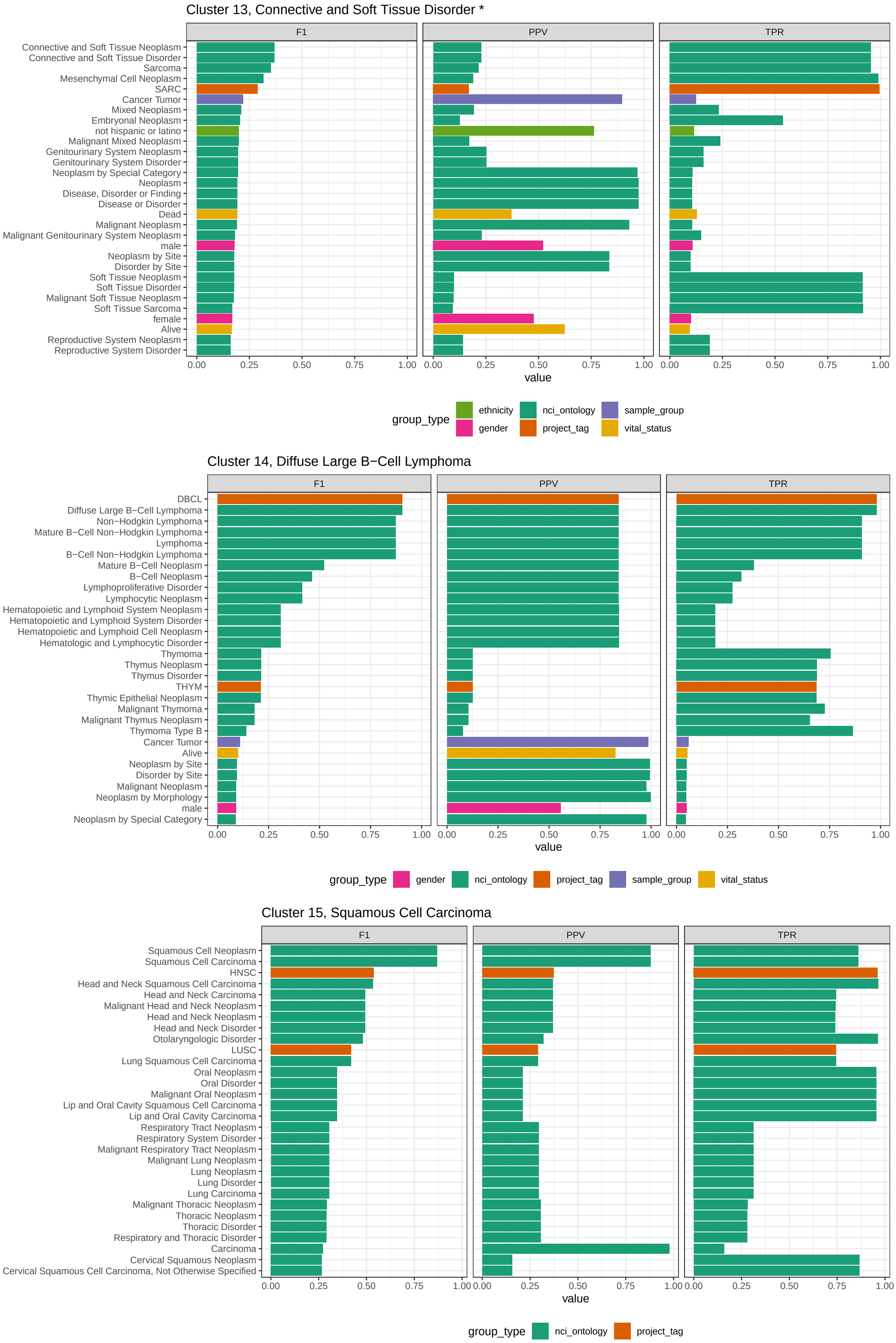


Figure 9: Prioritisation of NCI terms and other forms of meta-data used to annotate clusters 10–12. Cluster 11 contains mostly Lung Adenocarcinoma (LUAD), but also Lung Squamous Cell Carcinoma (LUSC), and thus should have been labelled Lung Carcinoma instead.



Cluster 13, Connective and Soft Tissue Disorder *

Cluster 14, Diffuse Large B−Cell Lymphoma

Cluster 15, Squamous Cell Carcinoma

Figure 10: Prioritisation of NCI terms and other forms of meta-data used to annotate clusters 13–15. Cluster 13 contains a mixture of cancer types, and should have been labelled 'Mixed'.

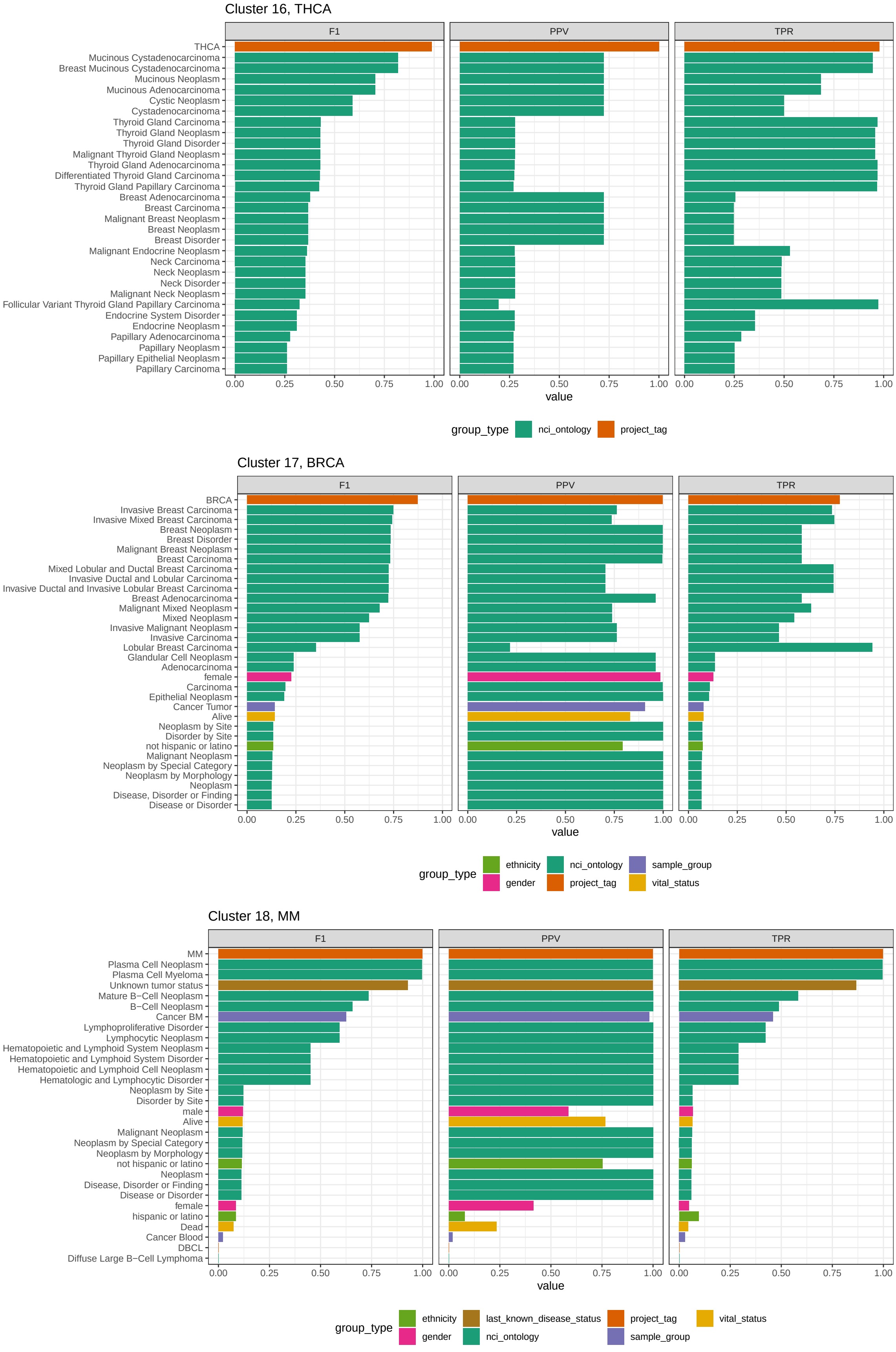


Figure 11: Prioritisation of NCI terms and other forms of meta-data used to annotate clusters 16–18.

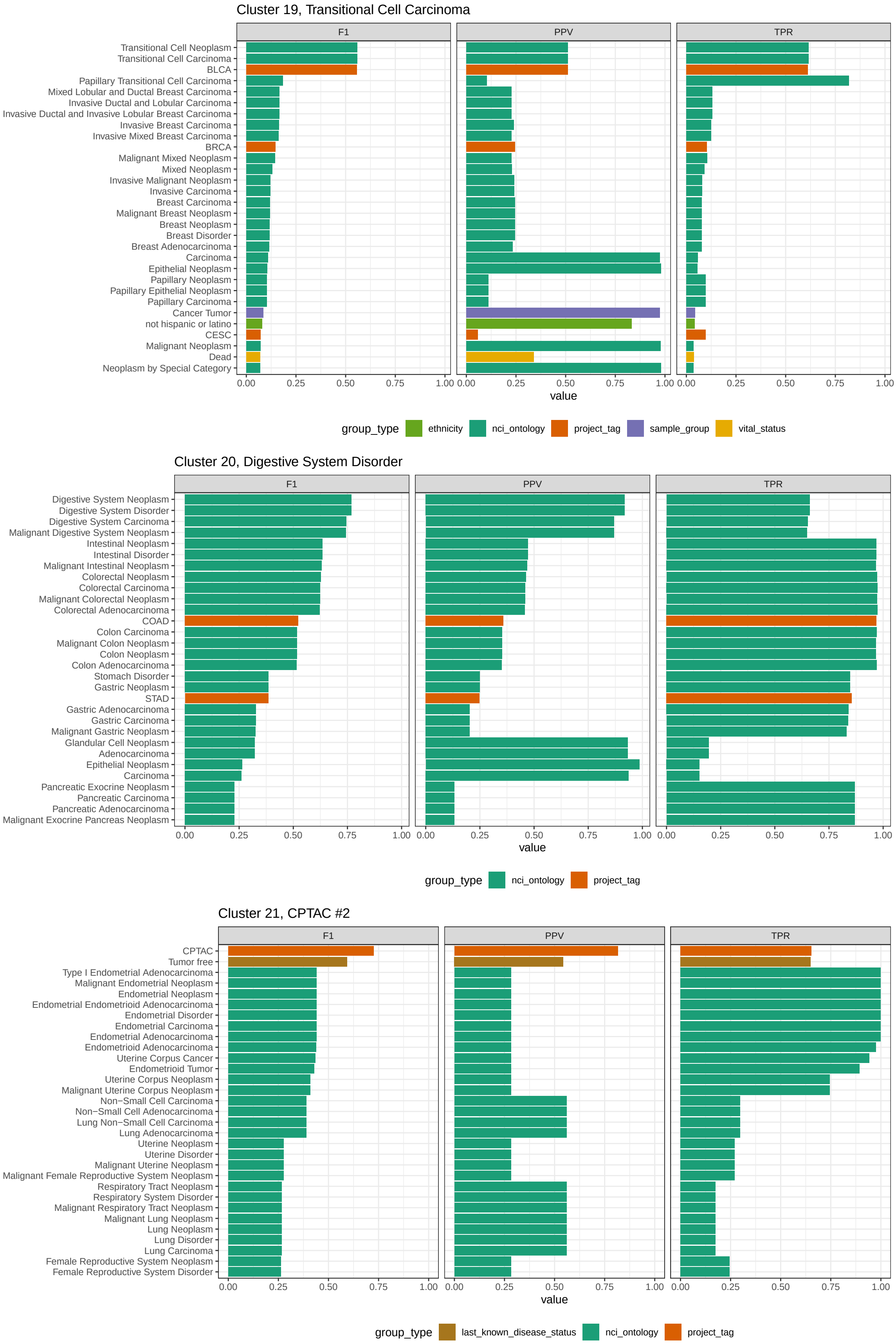


Figure 12: Prioritisation of NCI terms and other forms of meta-data used to annotate clusters 19–21. Cluster 21 contains a mixture of cancer types, mostly lung carcinoma and endometrial carcinoma.

5.1 Melanocytic neoplasm

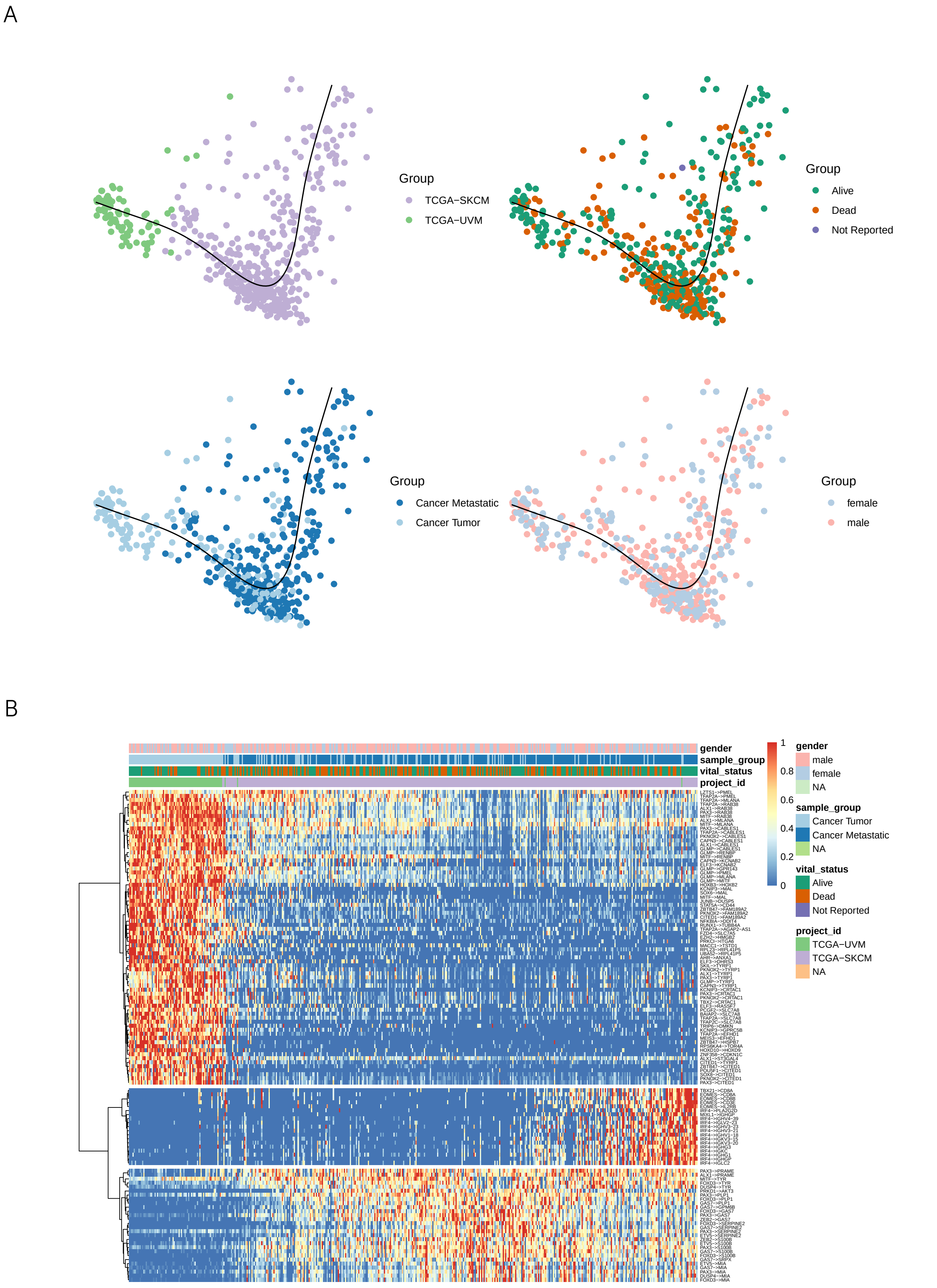


Figure 13: In-depth view of cluster 9, melanocytic neoplasm. A: A dimensionality reduction of the samples, coloured according to multiple sources of meta-data. **B:** The samples were ordered linearly with SCORPIUS (see trajectory in **A**) in order to visualise regulome activity in the form of a heatmap.

5.2 Kidney carcinoma

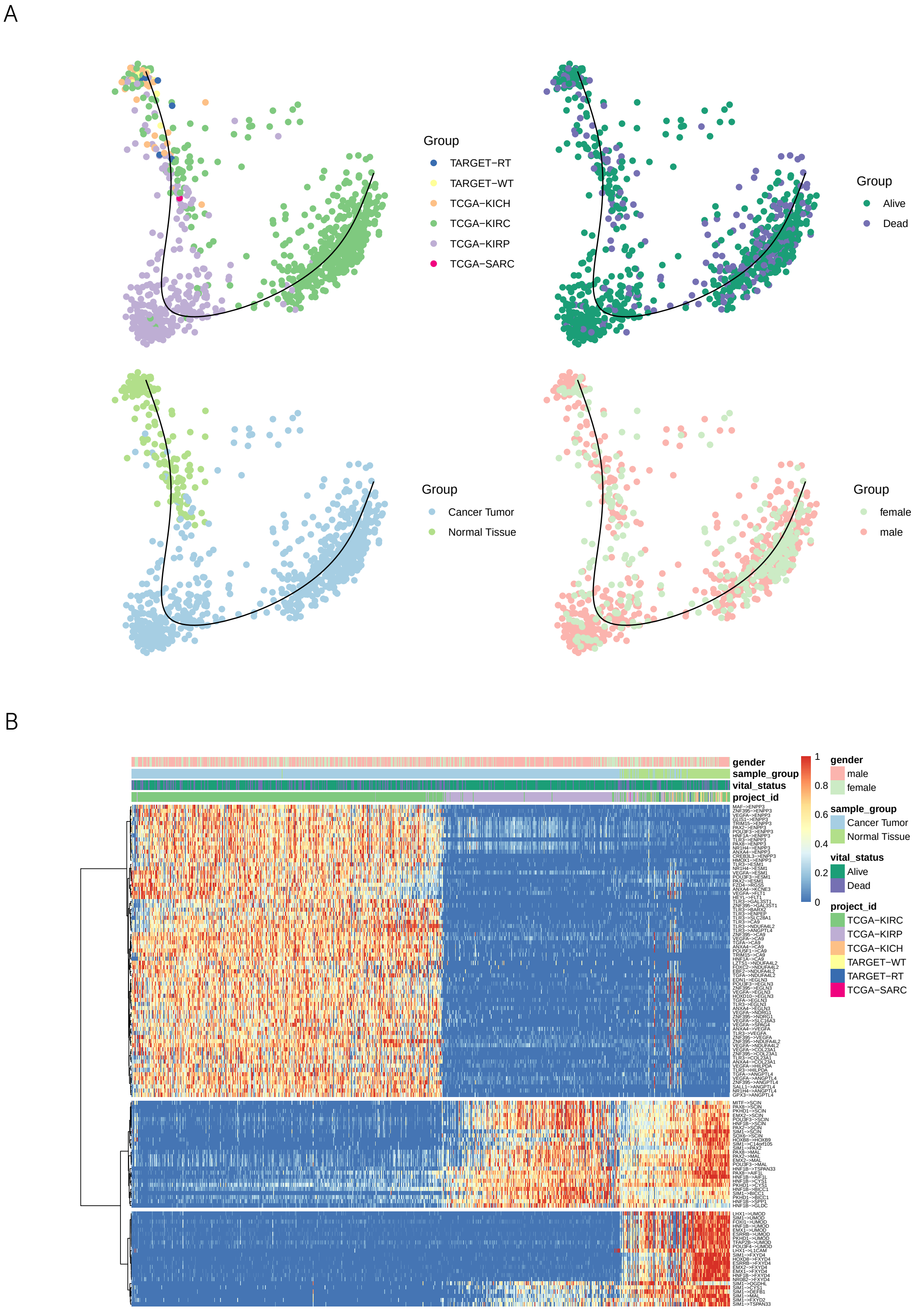


Figure 14: In-depth view of cluster 12, kidney carcinoma. A: A dimensionality reduction of the samples, coloured according to multiple sources of meta-data. **B:** The samples were ordered linearly with SCORPIUS (see trajectory in **A**) in order to visualise regulome activity in the form of a heatmap.

6 References

[1] Ziv Bar-Joseph, Anthony Gitter, and Itamar Simon. "Studying and Modelling Dynamic Biological Processes Using Time-Series Gene Expression Data". In: *Nat. Rev. Genet.* 13.8 (Aug. 2012), pp. 552–564.

[2] Noa Novershtern et al. "Densely Interconnected Transcriptional Circuits Control Cell States in Human Hematopoiesis". In: *Cell* 144.2 (2011), pp. 296–309.

[3] Gillian May et al. "Dynamic Analysis of Gene Expression and Genome-Wide Transcription Factor Binding during Lineage Specification of Multipotent Progenitors". In: *Cell Stem Cell* 13.6 (2013), pp. 754–768.

[4] Vladimir Jojic et al. "Identification of Transcriptional Regulators in the Mouse Immune System". In: *Nat. Immunol.* 14.6 (2013), pp. 633–643. DOI: 10.1038/ni.2587.Identification.

[5] Debbie K Goode et al. "Dynamic Gene Regulatory Networks Drive Hematopoietic Specification and Differentiation". In: *Dev. Cell* 36.5 (2016), pp. 572–587.

[6] Victoria Moignard et al. "Characterization of Transcriptional Networks in Blood Stem and Progenitor Cells Using High-Throughput Single-Cell Gene Expression Analysis". In: *Nat. Cell Biol.* 15.4 (Apr. 2013), pp. 363–372.

[7] Cristina Pina et al. "Single-Cell Network Analysis Identifies DDIT3 as a Nodal Lineage Regulator in Hematopoiesis." In: *Cell reports* 11.10 (2015), pp. 1503–1510. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2015.05.016. pmid: 26051941.

[8] Balázs Papp and Stephen Oliver. "Genome-Wide Analysis of the Context-Dependence of Regulatory Networks". In: *Genome Biology* 6.2 (Jan. 27, 2005), p. 206. ISSN: 1474-760X. DOI: 10.1186/gb-2005-6-2-206.

[9] Marieke Lydia Kuijjer et al. "Estimating Sample-Specific Regulatory Networks". In: *iScience* 14 (Mar. 28, 2019), pp. 226–240. ISSN: 2589-0042. DOI: 10.1016/j.isci.2019.03.021. pmid: 30981959.

[10] Xiaoping Liu et al. "Personalized Characterization of Diseases Using Sample-Specific Networks". In: *Nucleic Acids Research* 44.22 (2016), e164–e164. ISSN: 0305-1048. DOI: 10.1093/nar/gkw772. pmid: 27596597.

[11] Sara Aibar et al. "SCENIC: Single-Cell Regulatory Network Inference and Clustering". In: *Nature Methods* (Oct. 2017). ISSN: 1548-7091. DOI: 10.1038/nmeth.4463.

[12] Vân Anh Huynh-Thu et al. "Inferring Regulatory Networks from Expression Data Using Tree-Based Methods". In: *PLoS ONE* 5.9 (Jan. 2010), e12776. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0012776. pmid: 20927193.

[13] Leo Breiman. "Random Forests". In: *Machine Learning* 45 (2001), pp. 5–32.

[14] John N Weinstein et al. "The Cancer Genome Atlas Pan-Cancer Analysis Project." In: *Nature genetics* 45.10 (Oct. 2013), pp. 1113–20. ISSN: 1546-1718. DOI: 10.1038/ng.2764. pmid: 24071849.

[15] Thomas M. J. Fruchterman and Edward M. Reingold. "Graph Drawing by Force-Directed Placement". In: *Software: Practice and Experience* 21.11 (1991), pp. 1129–1164. ISSN: 1097-024X. DOI: `10.1002/spe.4380211102`.

[16] Vincent D Blondel et al. "Fast Unfolding of Communities in Large Networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 9, 2008), P10008. ISSN: 1742-5468. DOI: `10.1088/1742-5468/2008/10/p10008`.

[17] Nicholas Sioutos et al. "NCI Thesaurus: A Semantic Model Integrating Cancer-Related Clinical and Molecular Information". In: *Journal of Biomedical Informatics*. Bio*Medical Informatics 40.1 (Feb. 1, 2007), pp. 30–43. ISSN: 1532-0464. DOI: `10.1016/j.jbi.2006.02.013`.

[18] Nicholas Schaum et al. "Single-Cell Transcriptomics of 20 Mouse Organs Creates a Tabula Muris". In: *Nature* 562.7727 (Oct. 2018), pp. 367–372. ISSN: 1476-4687. DOI: `10.1038/s41586-018-0590-4`.

[19] Robrecht Cannoodt, Wouter Saelens, and Yvan Saeys. "Dyngen: Benchmarking with *in Silico* Single Cells". In: *In preparation* (Sept. 2019).

[20] L Breiman et al. *Classification and Regression Trees*. Wadsworth Publishing Company, 1984.

[21] Marvin N Wright and Andreas Ziegler. "Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R". In: *Journal of Statistical Software* 77.1 (Mar. 2017). DOI: `10.18637/jss.v077.i01`.

[22] Seunghak Lee and Seungjin Choi. "Landmark MDS Ensemble". In: *Pattern Recognition* 42.9 (Sept. 2009), pp. 2045–2053. ISSN: 00313203. DOI: `10.1016/j.patcog.2008.11.039`.

[23] Gabor Csardi and Tamas Nepusz. "The Igraph Software Package for Complex Network Research". In: *InterJournal, Complex Systems* 1695.5 (2006), pp. 1–9.

[24] Paul Shannon et al. "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks". In: *Genome Research* 13.11 (Nov. 1, 2003), pp. 2498–2504. DOI: `10.1101/gr.1239303`.