Supplementary Files

Upfront whole blood transcriptional patterns in patients receiving immune checkpoint inhibitors associate with clinical outcome

S. Hone Lopez[1], S. Loipfinger[1], A. Bhattacharya[1], M. Jalving[1], S.F. Oosting[1], T.J.N. Hiltermann[2], M. de Bruyn[3], E.G.E. de Vries[1], J.J. de Haan[1] & R.S.N. Fehrmann[1].

Departments of [1]Medical Oncology, [2]Pulmonary Diseases, and [3]Obstetrics and Gynecology, University Medical Center Groningen, University of Groningen, Groningen, The Netherlands.

S1. Supplementary Files

S2. Supplementary Data

Provided separately

Supplementary Data 1. ICA annotated file

Supplementary Data 2. Gene set enrichment analyses for all 1,262 TCs

Supplementary Data 3. Biological characterization the TCs associated with tumor response, PFS, and OS

Supplementary Data 4. Source Data for the Figures

Supplementary Methods

Acquisition of whole blood transcriptional profiles from public repositories: PUBLIC dataset

Public WB transcriptional profiles were acquired from the ARCHS4 database, which aggregates transcriptional profiles sourced from a range of platforms, including HiSeq 2000, HiSeq 2500, and NextSeq 500 platforms, and encompasses various species. This database compiles datasets from the Gene Expression Omnibus (GEO) and Sequence Read Archive (SRA) repositories. [1] For our analysis, we specifically sourced human gene-level data from ARCHS4 Version 2.2, corresponding to Ensembl release 107.

To ensure a diverse collection of WB transcriptional profiles, the human gene-level data from ARCHS4 was carefully curated. Only profiles annotated with 'paxgene' to specifically denote WB samples were included. Concurrently, we applied an automatic sample extraction for whole blood to make sure to exclude as much as possible the following terms: 'cancer', 'pbmc', 'tcell', 'monocyte', 'neutrophil', 'cd14', 'leukocyte', 'umbilical', 'skin', 'mononuclear', and 'white blood'. Quality control filters were applied to ensure dataset.

First, we retained only genes from coding sequences and genes with at least 20 reads in 1% of samples, eliminating non-functional pseudo-genes and genes with very low transcript abundance. We then removed samples with a gene count sum of less than 218 and zero counts in over 50% of all genes, excluding sparse and single-cell data. We removed samples with duplicated gene counts, retaining one randomly selected sample. A weak quality control step was performed using DESeq2 v1.30.1 size factor calculation with the "poscounts" estimator. [2] Samples with a size factor of <0.1 and >5 were removed to exclude samples with an overall too low or high median

gene count level. Additionally, PCA was conducted on the sample correlation matrix of the size factor normalized and log2(x+1) transformed gene expression profiles. Samples which failed the outlier detection threshold based on Tukey's rule on PCA's first principal component activity were removed using the tukey_mc_up(PC1, coef=1) function from the R package bigutilsr v0.3.4.

The transcriptional profiles were then normalized to account for variations in sequencing library size, employing size factors, followed by a $\log_2(x+1)$ transformation to stabilize the variance across the dataset. [3] To reduce the influence of technical variability attributable to sequencing platforms on the transcriptional profiles, principal component analysis (PCA) was employed on the correlation matrix of the profiles. This analysis allowed for identifying and removing the dominant non-biological variation, represented by the first principal component and primarily associated with platform-specific artifacts. This refined dataset is referred to as the PUBLIC dataset.

To annotate the gene expression data with relevant gene information and National Center for Biotechnology Information (NCBI) identifiers, the BioMart tool was used for cross-referencing Ensembl gene IDs. [2]

Consensus-independent component analysis and projection of PRIMERO profiles onto transcriptional components from the PUBLIC dataset

A) Whitening the bulk transcriptional profiles

Whitening the bulk transcriptional profiles is an essential preprocessing step before applying ICA, where we transform the matrix $\boldsymbol{X}$ containing the bulk transcriptional profiles in rows and measurements for p genes in the columns into a new matrix $\boldsymbol{X}_{whitened}$ with whitened transcriptional profiles. This transformation gives the whitened transcriptional profiles desirable statistical properties:

1) *Unit variance*: This standardizes the scale of each whitened transcriptional profile, ensuring that each has a variance of 1. As a result, all whitened transcriptional profiles are on a comparable scale, which is critical for the subsequent analysis because it prevents any single whitened transcriptional profile from dominating the results due to differences in the magnitude of gene expression.
2) *Zero covariance*: By transforming the bulk transcriptional profiles so that the covariance between any pair is zero, we ensure they are uncorrelated. In the covariance matrix of $\boldsymbol{X}_{whitened}$ this is reflected by having diagonal elements representing variances (now one due to unit variance scaling) and off-diagonal elements (representing covariances between different profiles) being zero.
3) *Orthogonality*: With zero covariance, the whitened transcriptional profiles also achieve orthogonality, meaning they are perpendicular to each other in the multidimensional gene expression space. This orthogonality provides a robust basis for separating the whitened transcriptional profiles into independent components during the ICA process.

The process of whitening is critical when performed before applying ICA to bulk transcriptional profiles for two main reasons:

1) *Enhancing the efficiency of ICA*: As described above, whitening transforms the matrix of bulk transcriptional profiles, denoted as $\boldsymbol{X}$, into a new matrix $\boldsymbol{X}_{whitened}$ with desirable statistical properties. Such a transformation simplifies the optimization landscape for ICA, making it smoother and more tractable. Consequently, this increases the efficiency of the ICA algorithm, leading to faster convergence.

2) *Reducing dimensionality and mitigating noise*: Beyond expediting convergence, whitening can play a role in reducing the dimensionality of the data and mitigating noise. By transforming the bulk transcriptional profiles into an orthogonal and uncorrelated space, it becomes feasible to pinpoint and disregard whitened transcriptional profiles that contribute minimal variance, which often represent noise or less informative transcriptional patterns in the data. Discarding these whitened transcriptional profiles reduces computational demand and enhances the ability of ICA to discern transcriptional patterns that reflect distinct biological processes.

Whitening the matrix $\boldsymbol{X}$ containing the bulk transcriptional profiles involves the following steps:

1) *Center each profile*: To center each bulk transcriptional profile in matrix $\boldsymbol{X}$, subtract the mean of its respective row. The centered matrix $\widetilde{\boldsymbol{X}}$ is obtained by the operation $\widetilde{\boldsymbol{X}} = \boldsymbol{X} - \mu 1^T$, where μ is the vector of row means, and 1^T is a row vector of ones with a length matching the number of columns in $\boldsymbol{X}$. This process adjusts each value in a row by the row's mean, centering the bulk transcriptional profiles.
2) Calculate the covariance matrix: Compute the covariance matrix $\boldsymbol{C}$ of the matrix $\widetilde{\boldsymbol{X}}$. The covariance is computed as $\boldsymbol{C} = \frac{1}{N-1}\widetilde{\boldsymbol{X}}\boldsymbol{X}^T$, where $\boldsymbol{X}^T$ is the transpose of $\widetilde{\boldsymbol{X}}$, and N is the number of genes (columns of $\widetilde{\boldsymbol{X}}$).
3) *Eigenvalue decomposition*: Perform an eigenvalue decomposition on the covariance matrix $\boldsymbol{C}$. Since $\boldsymbol{C}$ is a positive semi-definite matrix, it can be decomposed $\boldsymbol{C} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$, where $\boldsymbol{V}$ is a matrix of eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix with eigenvalues on the diagonal. The eigenvectors correspond

to the principal axes of the data distribution, and the eigenvalues correspond to the variance explained by each axis.

4) *Form the whitening transform*: Create the whitening transform matrix $\boldsymbol{W}$ using the eigenvalues and eigenvectors. Each eigenvalue λ_i is used to form a scaling factor $\lambda_i^{-1/2}$ (assuming λ_i is non-zero). The whitening matrix $\boldsymbol{W}$ is then constructed as $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}^T$, where $\boldsymbol{\Lambda}^{-1/2}$is the diagonal matrix of the inverse square roots of the eigenvalues.
5) *Apply the whitening transform*: Multiply the matrix $\widetilde{\boldsymbol{X}}$ by the whitening matrix $\boldsymbol{W}$ to obtain the whitened matrix $\boldsymbol{X}_{whitened} = \widetilde{\boldsymbol{X}}\boldsymbol{W}$. The result is that the covariance matrix of $\boldsymbol{X}_{whitened}$ is the identity matrix $\boldsymbol{I}$, meaning that the rows of $\boldsymbol{X}_{whitened}$ are linearly uncorrelated/orthogonal, and each has unit variance.
6) *Dimensionality and noise reduction (optional):* You can choose a subset of the eigenvectors—for example, corresponding to the largest eigenvalues—before constructing $\boldsymbol{W}$. This effectively reduces the number of dimensions in the whitened data, as you're only keeping the eigenvectors with the most variance, which typically carry the most information. This automatically discards the eigenvectors that contribute the least variance. These eigenvectors often represent noise or less informative features of the data.

In the whitening process, there's an important setting known as the *Cumulative Explained Variance Threshold*. This setting allows to specify the amount of the original data's variance we want to preserve in the whitened data. The cumulative explained variance is the total variance accounted for by the selected eigenvectors. Choosing a cumulative explained variance threshold—like 90%—determines that we want to keep enough eigenvectors to capture 90% of the original variance. In practice, we would

add up the eigenvalues from largest to smallest until the sum is equal to or just exceeds 90% of the sum of all eigenvalues. The minimum number of eigenvalues we add up to meet this threshold is the number we keep. The remaining eigenvectors, which correspond to smaller eigenvalues and thus less variance, are discarded. As described above, this approach ensures that the whitened matrix $\boldsymbol{X}_{whitened}$ maintains most of the informative variance from the original matrix while reducing dimensionality and potentially removing noise. This matrix $\boldsymbol{X}_{whitened}$ with whitened transcriptional profiles serves as input to ICA.

B) Applying Independent Component Analysis on the whitened transcriptional profiles

ICA is performed with the fastICA algorithm. FastICA is an iterative, fixed-point algorithm designed for the task of ICA. It reaches a solution through an iterative process that converges to a fixed point where the algorithm's output becomes stable and does not change significantly with further iterations. A detailed description of the fastICA algorithm has been provided by Hyvarinen et al. [4] In brief, the algorithm operates under the assumption that the observed bulk transcriptional profiles are linear mixtures of statistically independent, non-Gaussian source transcriptional patterns. The primary objective is to estimate the unmixing matrix that recovers these statistically independent source transcriptional patterns (i.e., TCs) when applied to the observed bulk transcriptional profiles.

Four key parameters govern FastICA:

1) *Mu* (μ): This parameter is often referred to as the learning rate or step size in the context of iterative optimization algorithms. In FastICA, *Mu* controls the magnitude of the updates made to the unmixing matrix in each iteration. An adequately chosen *Mu* is crucial for the algorithm's convergence; too large a

Mu may cause the algorithm to oscillate or diverge, while too small a *Mu* can lead to very slow convergence.

2) *Contrast Function* (g): The contrast function is a key component in ICA algorithms that measures non-Gaussianity. The contrast function maximizes the non-Gaussianity of the estimated source transcriptional patterns (i.e., TCs), assuming that the true source transcriptional patterns are statistically independent and have non-Gaussian distributions. Different choices of contrast functions can lead to different fastICA solutions, and the choice of function may depend on the nature of the true source transcriptional patterns. Common examples include hyperbolic tangent, kurtosis, raise to power 3, and skewness. The contrast function directly influences the update rule applied to the unmixing matrix during optimization.

3) *Epsilon* (ε): This parameter is a small positive value that serves as a convergence criterion. Specifically, *Epsilon* is used to determine when the change in the non-Gaussianity metric between consecutive iterations is small enough to consider the algorithm to have converged. In other words, when the algorithm's updates lead to changes smaller than *Epsilon*, the algorithm stops iterating, assuming that further iterations will not lead to significant improvements. This parameter helps to prevent infinite loops in cases where exact convergence to a threshold is not achievable.

4) *Maximum Number of Iterations* (max_{iter}): This parameter sets an upper limit on the number of iterations the algorithm will perform. This safeguards against non-convergence or extremely slow convergence, ensuring that the algorithm terminates after a reasonable time even if the *Epsilon* convergence criterion has not been met. The choice of this value may depend on the application

context and the computational resources available; a higher value allows for more thorough convergence at the cost of increased computational time.

FastICA employs a random initialization for the unmixing matrix and refines the unmixing matrix through an iterative process. Each iteration consists of two main steps: calculating the unmixing matrix update based on a chosen non-linear contrast function and the subsequent orthogonalization of the updated unmixing matrix to ensure that the estimated source transcriptional patterns remain statistically independent. The iterative process continues until a convergence criterion is met, defined as the change in the unmixing matrix between successive iterations falling below our predetermined threshold *Epsilon*. This criterion ensures that the algorithm terminates once the estimated source transcriptional patterns are sufficiently statistically independent and the unmixing matrix stabilizes. Upon convergence, the FastICA algorithm has decomposed the observed bulk transcriptional profiles into their constituent statistically independent source transcriptional patterns (i.e., TCs).

In formulas:

Given $\boldsymbol{X}_{whitened}$, which is the matrix containing the whitened transcriptional profiles, estimate the unmixing matrix $\boldsymbol{W}$ such that: $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}_{whitened}$. Here $\boldsymbol{S}$ is a matrix containing the independent source transcriptional patterns (i.e., TCs), which are assumed to be non-Gaussian and statistically independent. The FastICA algorithm proceeds iteratively as follows:

1) Initialization: Choose an initial random unmixing matrix $\boldsymbol{W}$.
2) Iteration:
    a. For each component i, update $\boldsymbol{W}$ using the fixed-point iteration scheme:

$$w_i^{new} = w_i + \mu(\mathbb{E}[\boldsymbol{X}_{whitened} \cdot g(w_i^T \boldsymbol{X}_{whitened})] - \mathbb{E}[g'(w_i^T \boldsymbol{X}_{whitened})] \cdot w_i)$$

Where:

- w_i is the i^{th} row of $\boldsymbol{W}$.
- $g(\cdot)$ is the non-linear contrast function and $g'(\cdot)$ its derivate.
- $\mathbb{E}[\cdot]$ denotes the expected value (mean).

3) Normalization: Normalize w_i^{new}: $w_i^{new} = \frac{w_i^{new}}{\|w_i^{new}\|}$.
4) Orthogonalization: Ensure that the components remain linearly uncorrelated by orthogonalizing the vectors: $\boldsymbol{W}^{new} = orthogonalize(\boldsymbol{W}^{new})$.
5) Convergence and iteration check: Check if the change in $\boldsymbol{W}$ is below the threshold ε. If it is, assume convergence: $\| \boldsymbol{W}^{new} - \boldsymbol{W} \| < \varepsilon$. If the convergence criterion is not met, set $\boldsymbol{W} = \boldsymbol{W}^{new}$ and repeat the iteration process if max_{iter} is not reached yet.
6) End: Once convergence is reached, the final unmixing matrix $\boldsymbol{W}$ can be used to compute the statistically independent components $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}_{whitened}$.

C) Consensus approach

The fastICA algorithm employs an optimization technique to decompose whitened transcriptional profiles of mixed signals $\boldsymbol{X}_{whitened}$ into statistically independent source transcriptional patterns (i.e., TCs) $\boldsymbol{S}$. However, this optimization process can converge to local optima, particularly in high-dimensional or noisy data spaces, leading to solutions that are not globally optimal. These local solutions may vary with different random initializations of the unmixing matrix $\boldsymbol{W}$, resulting in different sets of TCs $\boldsymbol{S}_{run}$ upon each run of the algorithm.

A consensus approach addresses this challenge by aggregating the results from multiple runs of the fastICA algorithm, each with a different random initialized

unmixing matrix $\boldsymbol{W}$ and identifying the TCs extracted across multiple runs. This approach aims to filter out TCs likely to be noise artifacts specific to a particular run or from convergence to a suboptimal local solution. The consensus approach enhances reliability by focusing on the stable and robust TCs that emerge across multiple runs.

Three parameters govern the consensus approach:

1) *Consensus Runs*: This parameter dictates how often the FastICA algorithm will be executed. In each run, the unmixing matrix $\boldsymbol{W}$ is randomly initialized with a different random seed, affecting the optimization process's starting conditions. The significance of this parameter lies in its ability to capture a wide range of potential solutions, which helps determine the consistency of the TCs identified across runs. A higher number of runs increases the chances of capturing stable and robust TCs, allowing for a more comprehensive sampling of the solution space. However, this also increases computational load.
2) *Consensus threshold*: This parameter specifies the minimum Pearson correlation coefficient TCs must exceed to be considered equivalent across different runs. A higher correlation threshold requires a stronger linear relationship between TCs to be regarded as the same. This could lead to identifying only the most consistent and robust TCs across runs. However, this might also dismiss less correlated but potentially relevant TCs, thereby reducing the sensitivity of the analysis.
3) *Credibility index*: This parameter serves as a threshold to determine whether a consensus-independent component (c-TCs) is robust enough to be considered reliable. It measures how consistently an TC appears across different runs. If a TC is identified in a proportion of runs that exceeds the credibility index, it is

deemed robust. The credibility index filters out TCs that are less likely to be true signals and more likely to be noise or artifacts, as these would not consistently appear across multiple runs. The credibility index's choice impacts the filtering process's strictness: setting it too high might exclude genuine TCs. At the same time, a too low threshold might include spurious ones.

The consensus approach involves several steps:

1) Consolidation of TCs: Initially, the TCs $\boldsymbol{S}_{run}$ derived from each run are combined to form a matrix $\boldsymbol{S}_{combined}$.
2) Computation of pair-wise correlations: Begin by centering each TC S in $\boldsymbol{S}_{combined}$ subtracting its mean and normalizing it to its L2-norm resulting in the matrix $\tilde{\boldsymbol{S}}_{combined}$. Then, calculate the product of the matrix $\tilde{\boldsymbol{S}}_{combined}$ with its transpose: $\tilde{\boldsymbol{S}}_{combined}(\tilde{\boldsymbol{S}}_{combined})^T$. The resulting matrix $\boldsymbol{R}$ will have diagonal elements that signify the self-correlation of each TC S, and off-diagonal elements that indicate the Pearson correlation coefficients between different TCs. The absolute value of these coefficients is then used to evaluate the similarity between the TCs.
3) Establishment of sets of linked TCs: For each TC S in the matrix $\boldsymbol{S}_{combined}$, the correlation matrix $\boldsymbol{R}$ is utilized to find TCs correlated with S above the *Consensus Threshold*. Those that exceed this threshold are deemed equivalent. TCs S and these equivalent TCs constitute a set of linked TCs L_{set}. Subsequently, TC S and its corresponding set L_{set} are paired as key-value and added to $\boldsymbol{L}_{map}$, where S serves as the key and L_{set} as the value.
4) Organize TCs into consensus clusters: An iterative process I for forming consensus clusters is started by processing $\boldsymbol{L}_{map}$:

- Take the largest set of linked TCs $L_{set_{largest}}$ from $\boldsymbol{L}_{map}$ and use its TCs to create a new consensus cluster.
- Then, a recursive process P starts to update $\boldsymbol{L}_{map}$ that takes a set of linked V L_{set} as parameter $Lparam$. This recursive process is started with $L_{set_{largest}}$ as parameter $Lparam$. In this recursive process P the following steps are performed:
    1. Eliminate any key-value pairs from $\boldsymbol{L}_{map}$ where the key is identical to any TC S found in $Lparam$.
    2. From the $\boldsymbol{L}_{map}$, for each pair that remains, update the pair's set of linked TCs L_{set}, by excluding any TC S that is included in $Lparam$.
    3. For each TC S within $Lparam$, we search $\boldsymbol{L}_{map}$ for a corresponding key that matches S. If no matching key is found, the current recursive process halts. Otherwise, if a matching pair is identified, we start another recursive process P, this time using the linked TC set L_{set} from the matching pair as the new $Lparam$.
- The iterative process I is stopped when there are no key-value pairs left in $\boldsymbol{L}_{map}$. Otherwise, another iteration of I is started with the current $L_{set_{largest}}$ from $\boldsymbol{L}_{map}$.

5) Filtering consensus clusters based on credibility index: We iterate over the consensus clusters, computing the credibility index for each. This index represents the ratio of the number of TCs within the clusters to the total number of runs (*consensus runs*). Clusters with a credibility index meeting or exceeding the *consensus threshold* are considered valid; the rest are removed. Suppose

316 the total number of valid consensus clusters surpasses the number of whitened
317 variables. In that case, we only retain the N clusters with the largest credibility
318 index, where N equals the number of whitened variables.
319 6) Determining c- TCs: For all TCs within a consensus cluster that persisted after
320 the credibility index-based filtering, we compute a correlation matrix as
321 described in *2) Computation of pair-wise correlations*. Then, for each
322 consensus cluster, we identify the most representative TC (the "winner") by
323 identifying the TC with the lowest average correlation to those outside its cluster.
324 We refer to this winner as the c- TC.
325 7) Flipping based on skewness: We determine the skewness of the gene weights
326 for each TC. If the skewness is negative, we flip all signs of the gene weights.

327 *C) Calculating the consensus mixing matrix*

328 As described above, ICA operates under the assumption that the observed bulk
329 transcriptional profiles can be understood as linear mixtures of source transcriptional
330 patterns. These mixtures are composed of statistically independent, non-Gaussian
331 source transcriptional patterns, which are encapsulated by what we refer to as c- TCs.
332 Consequently, each bulk transcriptional profile observed can be considered a
333 composite, where the c- TCs contribute with varying weights. These weights are
334 systematically arranged in what is known as a consensus mixing matrix, denoted by
335 $\boldsymbol{MM}_{consensus}$. In the $\boldsymbol{MM}_{consensus}$, each weight $\boldsymbol{MM}_{consensus_{(S_{consensus},X)}}$ represents the
336 'activity' of a c-TC $S_{consensus}$ in a bulk transcriptional profile X. In practical terms, to
337 derive the $\boldsymbol{MM}_{consensus}$ one must take the matrix $\boldsymbol{X}$, which contains the bulk
338 transcriptional profiles, and perform a matrix multiplication with the pseudo-inverse of
339 the c-TCs $\boldsymbol{S}_{consensus}$. This operation is mathematically represented as $\boldsymbol{MM}_{consensus}$=
340 $\boldsymbol{X}(\boldsymbol{S}_{consensus})^{-1}$.

D) Normalizing the consensus mixing matrix

In the analysis of bulk transcriptional profiles, we observe considerable variation in gene expression levels. This variation can often be attributed to technical factors such as platform inconsistencies or batch effects, rather than underlying biological differences. Additionally, within each c-TC, the number of genes carrying heigh weights varies, which could introduce bias into the calculated consensus mixing matrix, $\boldsymbol{MM}_{consensus}$. Such biases hamper direct comparisons within the $\boldsymbol{MM}_{consensus}$, whether comparing values from different profiles for a single component or across multiple components for a single profile. To address these challenges and facilitate more accurate comparisons, we have implemented a normalization methodology for the $\boldsymbol{MM}_{consensus}$:

1) Randomly permute the rows (containing genes) within the pseudo-inverse of the consensus-independent component to get $\left(\boldsymbol{S}_{consensus_{permuted}}\right)^{-1}$ and compute $\boldsymbol{MM}_{consensus_{permuted}} = \boldsymbol{X}\left(\boldsymbol{S}_{consensus_{permuted}}\right)^{-1}$. For each weight $\boldsymbol{MM}_{consensus_{permuted_{(S_{consensus},X)}}}$, add the weight to the null distribution $null\ distribution_{(S_{consensus},X)}$. Repeat this process for a specified number of permutations.
2) For every null distribution $null\ distribution_{(S_{consensus},X)}$, if the null distribution does not adhere to a Gaussian distribution, as determined by the Anderson-Darling test, we apply the Johnson transformation to convert the null distribution into a Gaussian distribution $null\ distribution_{Jtransformed_{(S_{consensus},X)}}$. This transformation employs one of three optimal families of distributions (S, SU, SL) and finds the parameters that transform the null distribution as much as

possible to a Gaussian distribution. Apply this transformation also to the corresponding $\boldsymbol{MM}_{consensus_{(S_{consensus},X)}}$ to get $\boldsymbol{MM}_{consensus_{Jtransformed_{(S_{consensus},X)}}}$.

3) We then fit a symmetrical, generalized Gaussian distribution to $null\ distribution_{Jtransformed_{(S_{consensus},X)}}$. Next, we extract the p-value of the $\boldsymbol{MM}_{consensus_{transformed_{(S_{consensus},X)}}}$ and convert it to a Z-score to obtain the final normalized mixing matrix weight $\boldsymbol{MM}_{normalized_{(S_{consensus},X)}}$.

Supplementary References

1. Lachmann A, Torre D, Keenan, AB, *et al.* Massive mining of publicly available RNA-seq data from human and mouse. *Nat Commun* **2018**;9:1366. Lachmann A, Torre D, Keenan AB, *et al.* Massive mining of publicly available RNA-seq data from human and mouse. *Nat Commun* **2018**;9:1366.
2. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* **2014**;15:550.
3. Durinck S, Spellman P, Birney E., Huber W. Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. *Nat Protoc* **2019**;4:1184–91.
4. Hyvärinen A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans Neural Netw* **1999**;10, 626–f34.

Supplementary Table 1. Co-treatment up to 3 months before, during, or up to 3 months after treatment with immune checkpoint inhibitors in the PRIMERO-cohort.

	CSCC (N=10)	Melanoma (N=83)	MCC (N=7)	NSCLC (N=17)	RCC (N=28)	Total (N=145)
Targeted therapy up to 3 months before ICIs						
VEGF inhibitor & IFN	0 (0%)	0 (0%)	0 (0%)	0 (0%)	4 (14.3%)	4 (2.8%)
BRAF MEK inhibitors	0 (0%)	24 (28.9%)	0 (0%)	0 (0%)	0 (0%)	24 (16.6%)
Capmatinib	0 (0%)	0 (0%)	0 (0%)	1 (5.9%)	0 (0%)	1 (0.7%)
TKI	0 (0%)	0 (0%)	0 (0%)	0 (0%)	4 (14.3%)	4 (2.8%)
None	10 (100%)	59 (71.1%)	7 (100%)	16 (94.1%)	20 (71.4%)	112 (77.2%)
Targeted therapy during ICIs						
TKI	0 (0%)	0 (0%)	0 (0%)	0 (0%)	4 (14.3%)	4 (2.8%)
VEGF inhibitor	0 (0%)	0 (0%)	0 (0%)	1 (5.9%)	0 (0%)	1 (0.7%)
None	10 (100%)	83 (100%)	7 (100%)	16 (94.1%)	24 (85.7%)	140 (96.6%)

	CSCC (N=10)	Melanoma (N=83)	MCC (N=7)	NSCLC (N=17)	RCC (N=28)	Total (N=145)
Targeted therapy up to 3 months after ICIs						
TKI	0 (0%)	0 (0%)	0 (0%)	0 (0%)	6 (21.4%)	6 (4.1%)
VEGF inhibitor	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (3.6%)	1 (0.7%)
BRAF MEK inhibitors	0 (0%)	12 (14.5%)	0 (0%)	0 (0%)	0 (0%)	12 (8.3%)
None	10 (100%)	71 (85.5%)	7 (100%)	17 (100%)	21 (75.0%)	126 (86.9%)
Chemotherapy up to 3 months before ICIs						
Yes	0 (0%)	1 (1.2%)	1 (14.3%)	0 (0%)	0 (0%)	2 (1.4%)
No	10 (100%)	82 (98.8%)	6 (85.7%)	17 (100%)	28 (100%)	143 (98.6%)
Chemotherapy during ICIs						
Yes	0 (0%)	0 (0%)	0 (0%)	10 (58.8%)	0 (0%)	10 (6.9%)
No	10 (100%)	83 (100%)	7 (100%)	7 (41.2%)	28 (100%)	135 (93.1%)
Chemotherapy up to 3 months after ICIs						
Yes	0 (0%)	1 (1.2%)	0 (0%)	4 (23.5%)	0 (0%)	5 (3.4%)

	CSCC (N=10)	Melanoma (N=83)	MCC (N=7)	NSCLC (N=17)	RCC (N=28)	Total (N=145)
No	10 (100%)	82 (98.8%)	7 (100%)	13 (76.5%)	28 (100%)	140 (96.6%)
Other therapies up to 3 months before ICIs						
TILs and IL-2	0 (0%)	1 (1.2%)	0 (0%)	0 (0%)	0 (0%)	1 (0.7%)
None	10 (100%)	82 (98.8%)	7 (100%)	17 (100%)	28 (100%)	144 (99.3%)
Other therapies up to 3 months after ICIs						
TILs and IL-2	0 (0%)	1 (1.2%)	0 (0%)	0 (0%)	0 (0%)	1 (0.7%)
TVEC	0 (0%)	1 (1.2%)	0 (0%)	0 (0%)	0 (0%)	1 (0.7%)
None	10 (100%)	81 (97.6%)	7 (100%)	17 (100%)	28 (100%)	143 (98.6%)
Radiotherapy up to 3 months before ICIs						
Yes	1 (10.0%)	17 (20.5%)	0 (0%)	4 (23.5%)	6 (21.4%)	28 (19.3%)
No	9 (90.0%)	66 (79.5%)	7 (100%)	13 (76.5%)	22 (78.6%)	117 (80.7%)
Radiotherapy during ICIs						
Yes	0 (0%)	11 (13.3%)	0 (0%)	1 (5.9%)	6 (21.4%)	18 (12.4%)

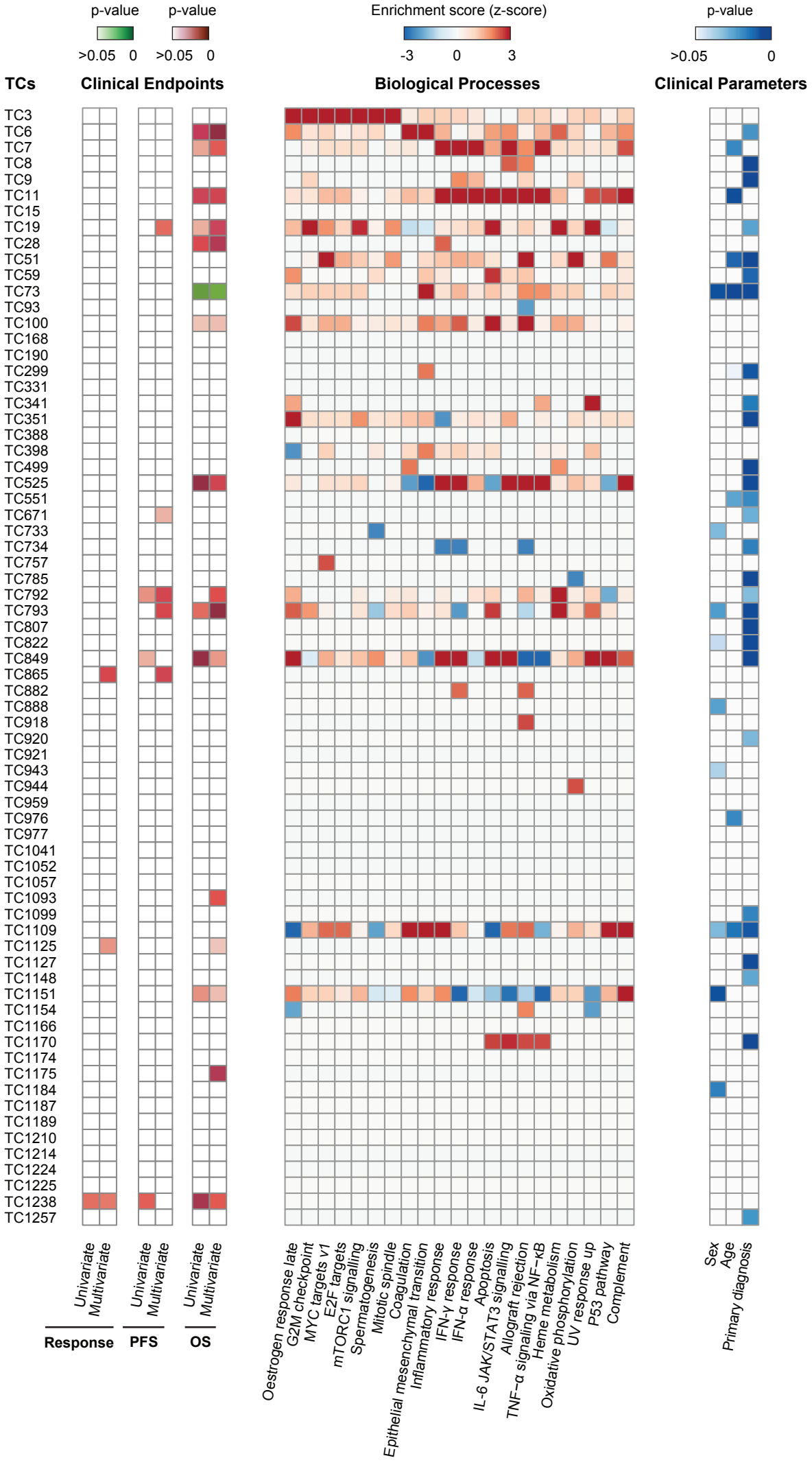
	CSCC (N=10)	Melanoma (N=83)	MCC (N=7)	NSCLC (N=17)	RCC (N=28)	Total (N=145)
No	10 (100%)	72 (86.7%)	7 (100%)	16 (94.1%)	22 (78.6%)	127 (87.6%)
Radiotherapy up to 3 months after ICIs						
Yes	0 (0%)	1 (1.2%)	0 (0%)	4 (23.5%)	0 (0%)	5 (3.4%)
No	10 (100%)	82 (98.8%)	7 (100%)	13 (76.5%)	28 (100%)	140 (96.6%)
Cancer surgery up to 3 months before ICIs						
Yes	1 (10.0%)	7 (8.4%)	0 (0%)	0 (0%)	0 (0%)	8 (5.5%)
No	9 (90.0%)	76 (91.6%)	7 (100%)	17 (100%)	28 (100%)	137 (94.5%)
Cancer surgery during ICIs						
Yes	0 (0%)	0 (0%)	0 (0%)	0 (0%)	4 (14.3%)	4 (2.8%)
No	10 (100%)	83 (100%)	7 (100%)	17 (100%)	24 (85.7%)	141 (97.2%)
Cancer surgery up to 3 months after ICIs						
Yes	1 (10.0%)	3 (3.6%)	0 (0%)	0 (0%)	2 (7.1%)	6 (4.1%)
No	9 (90.0%)	80 (96.4%)	7 (100%)	17 (100%)	26 (92.9%)	139 (95.9%)

CSCC, Cutaneous Squamous Cell Carcinoma; IL-2, Interleukin 2; ICIs, Immune Checkpoint Inhibitors; MCC, Merkel Cell Carcinoma; N, sample size; NSCLC, Non-Small Cell Lung Cancer; RCC, Renal Cell Carcinoma; SD, Standard Deviation, TILs, Tumor-Infiltrating Lymphocytes; TKI, Tyrosine kinase inhibitor; T-VEC, Talimogene laherparepve; VEGF inhibitor, Vascular Endothelial Growth Factor.

Supplementary Figures

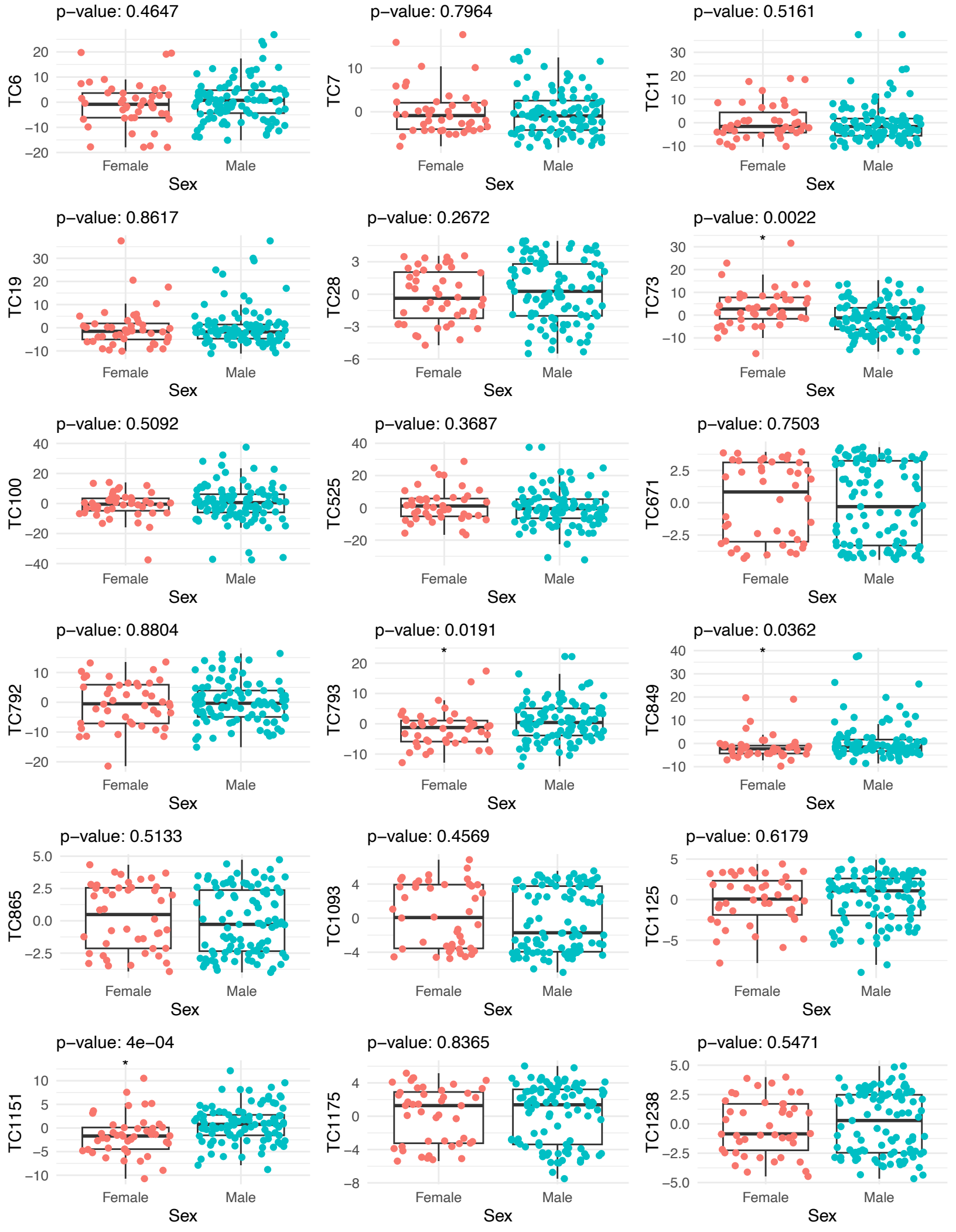
Supplementary Figure 1. Biology of the 70 TCs with a median absolute deviation score >3. The heatmaps in the left section illustrate the associations between TCs and OS, PFS, and tumor response to ICIs. TCs for which higher activity is associated with better clinical outcomes are indicated in green, while those associated with worse clinical outcomes are indicated in red. The shade of green or red reflects the significance of the association. The central heatmap displays the results of gene set enrichment analyses focusing on Hallmark gene sets. The heatmap on the right section of the figure shows the association between TC activity and clinical parameters.

Supplementary Figure 1

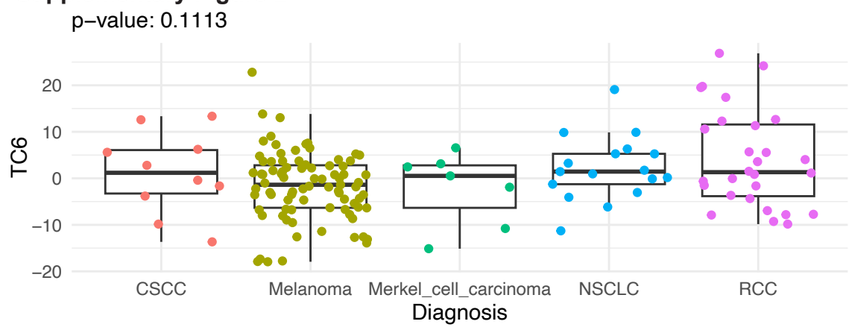


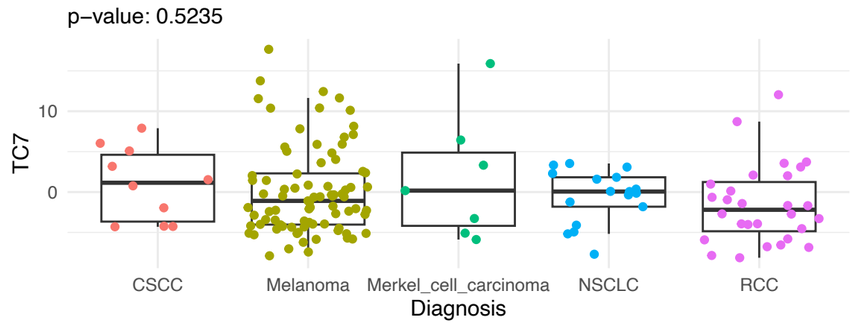
Supplementary Figure 2. **Mixing matrix activity differences for sex, age, and tumor type of outcome-associated TCs**. For each of the outcome-associated TCs, its activity's significant association with sex, age, and tumor type was determined. P-values for age were calculated using Spearman's rank correlation. P-values for sex and tumor type were calculated using multinomial log-linear models.

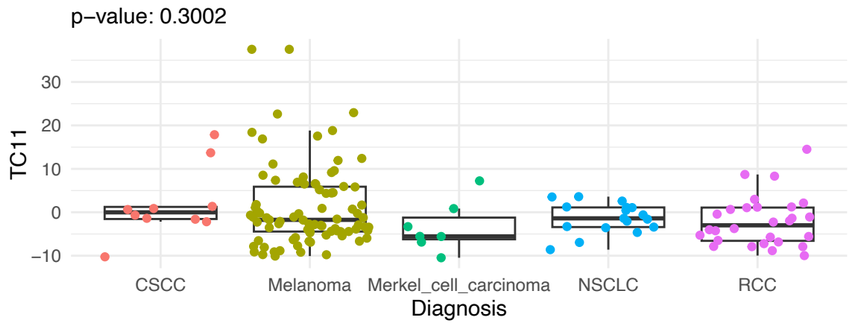
Supplementary Figure 2

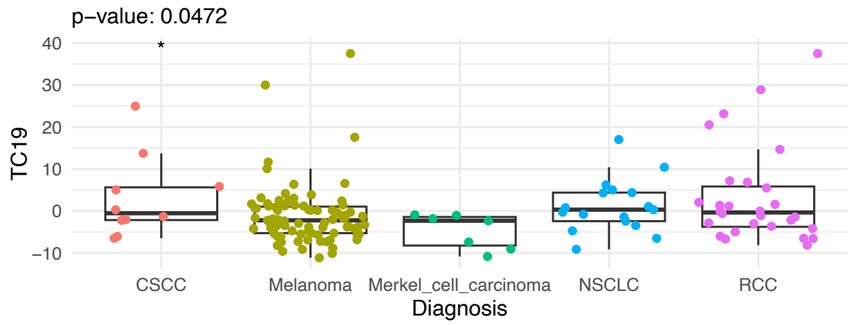


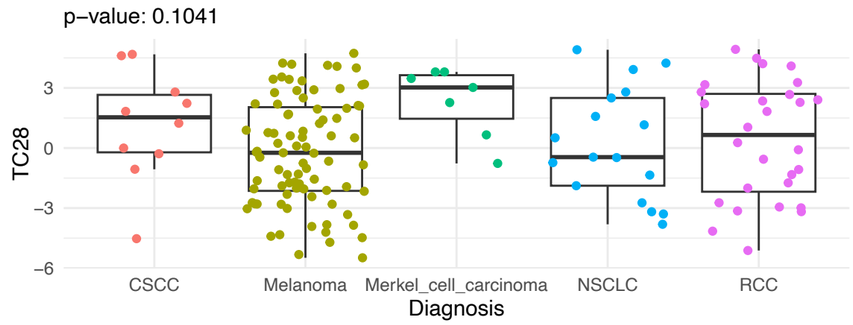
Supplementary Figure 2

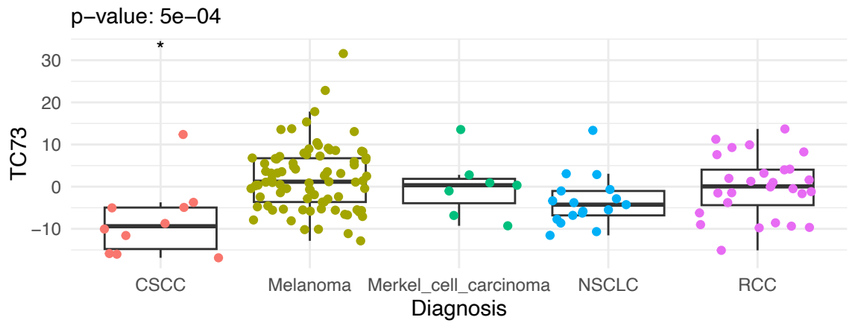


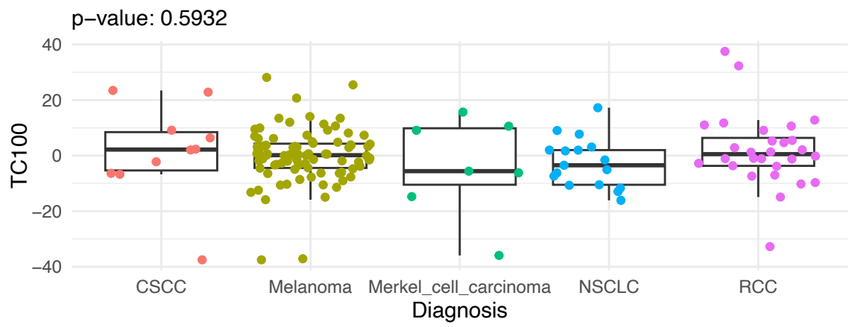


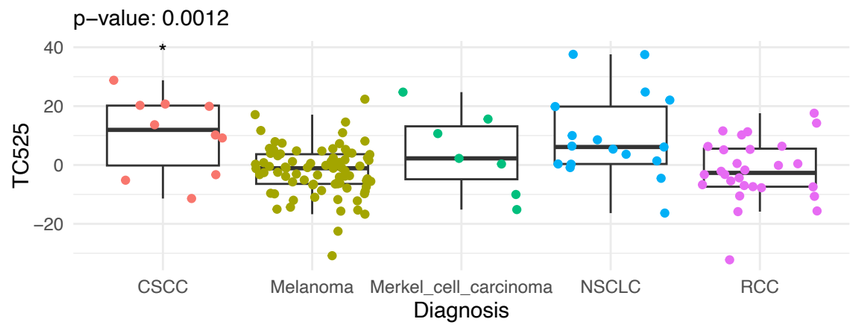


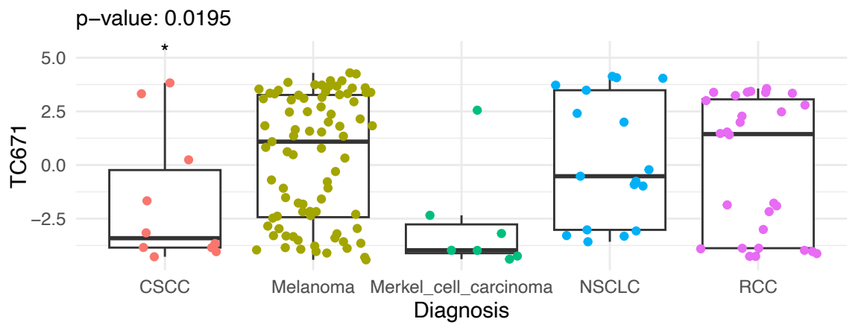


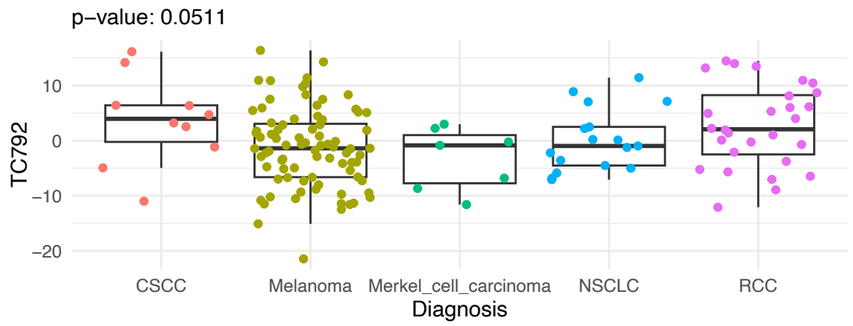


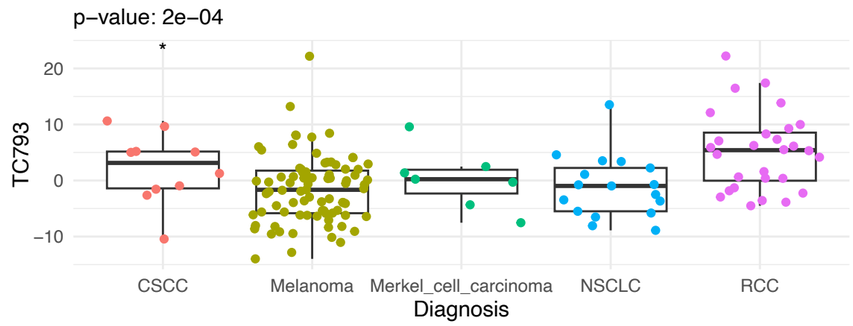


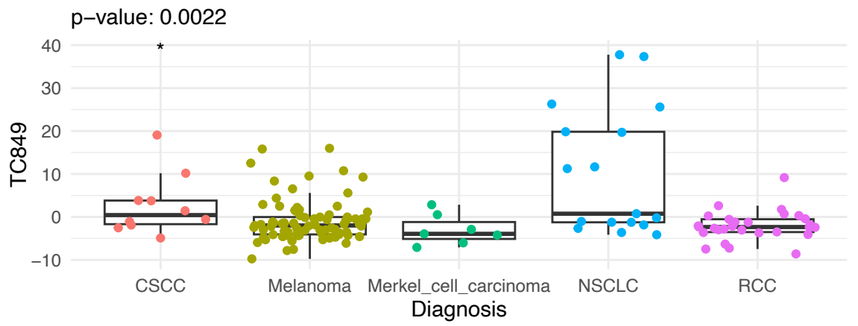


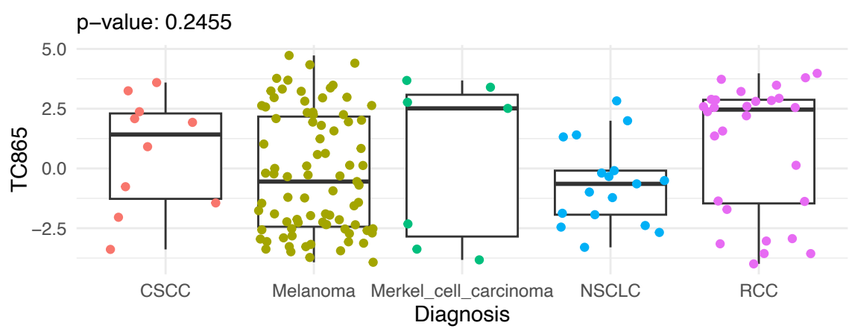


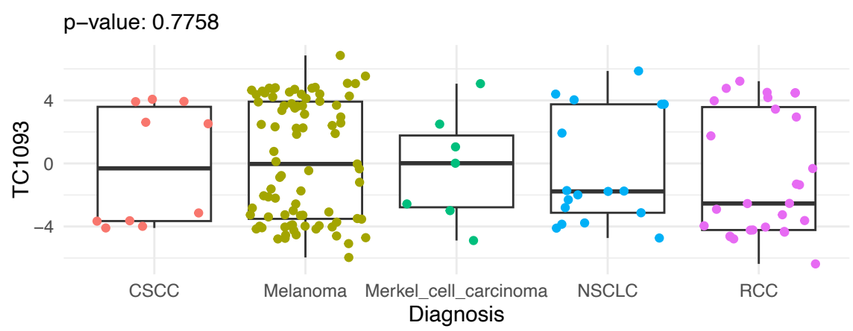


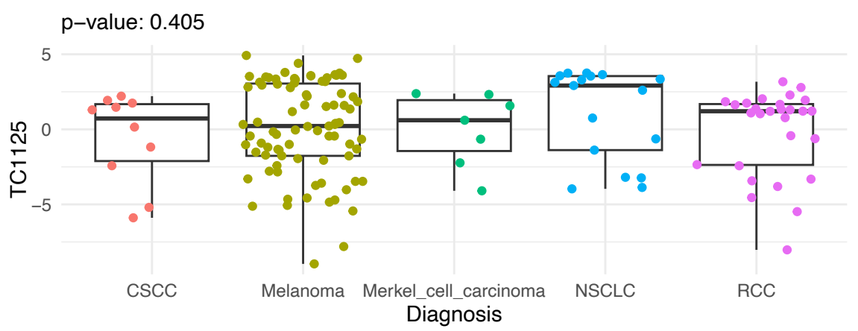


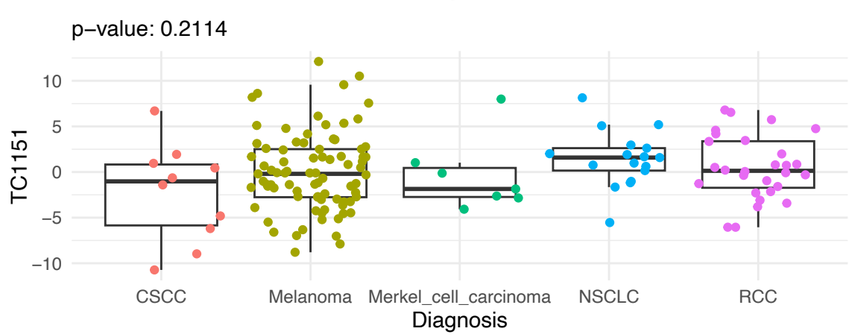


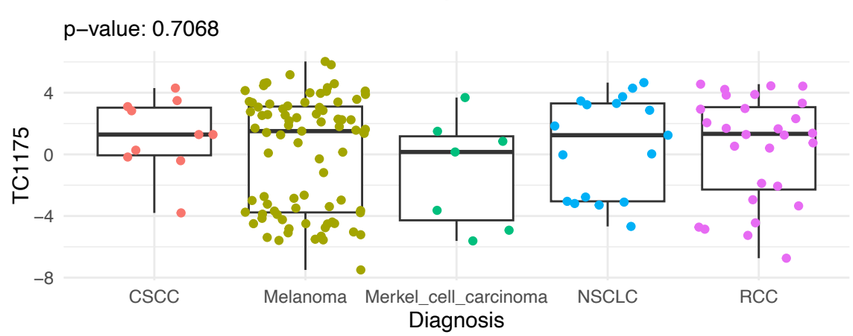


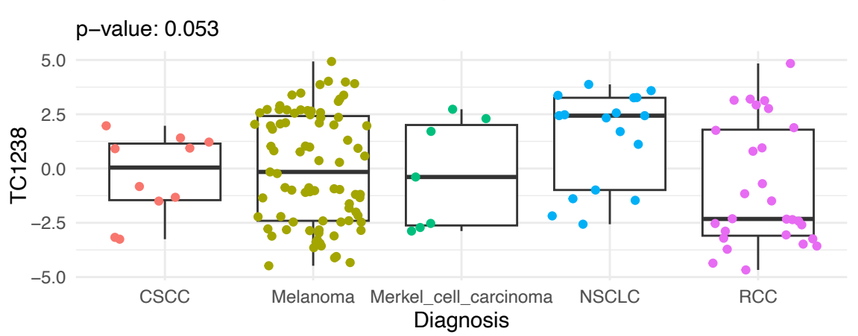




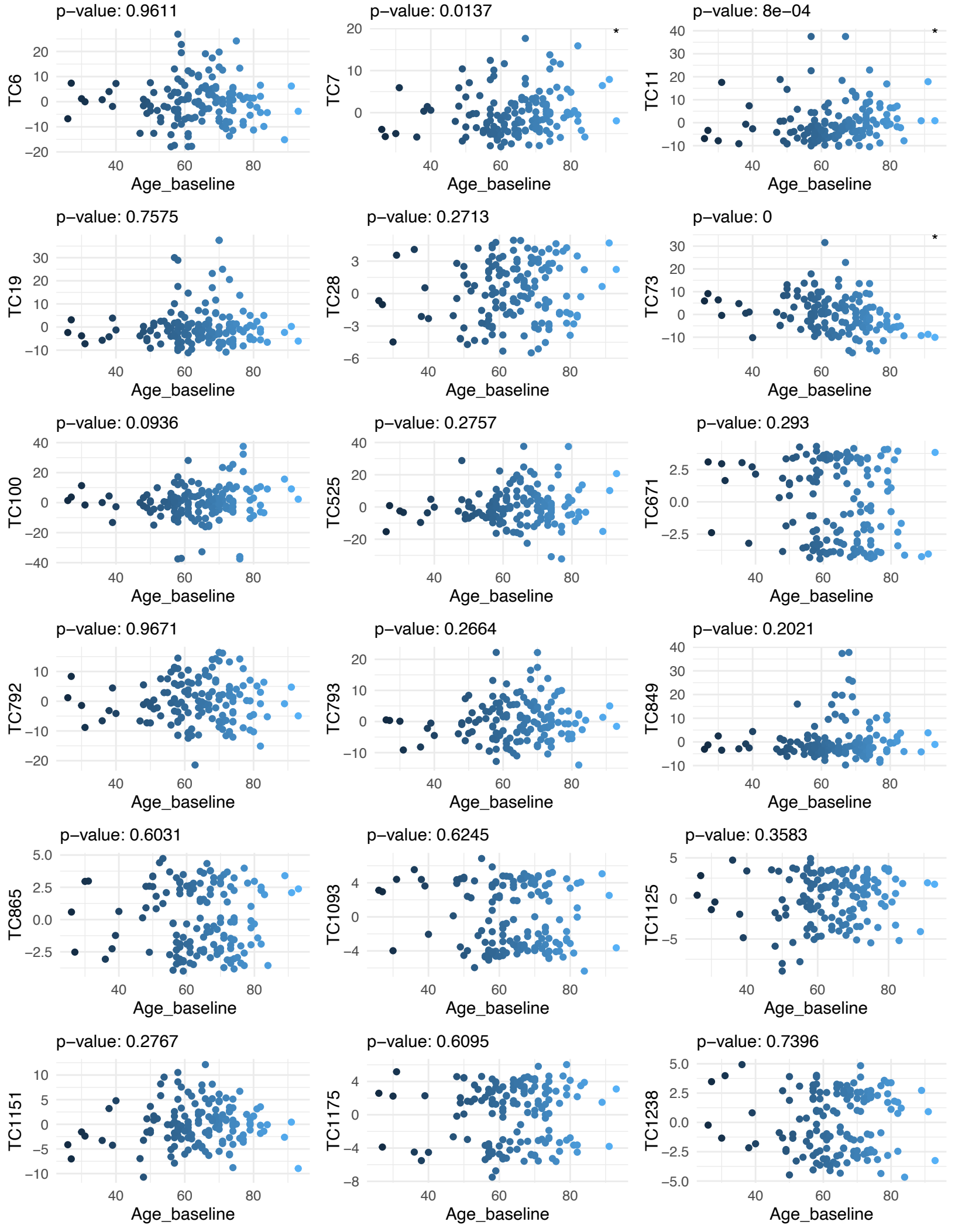






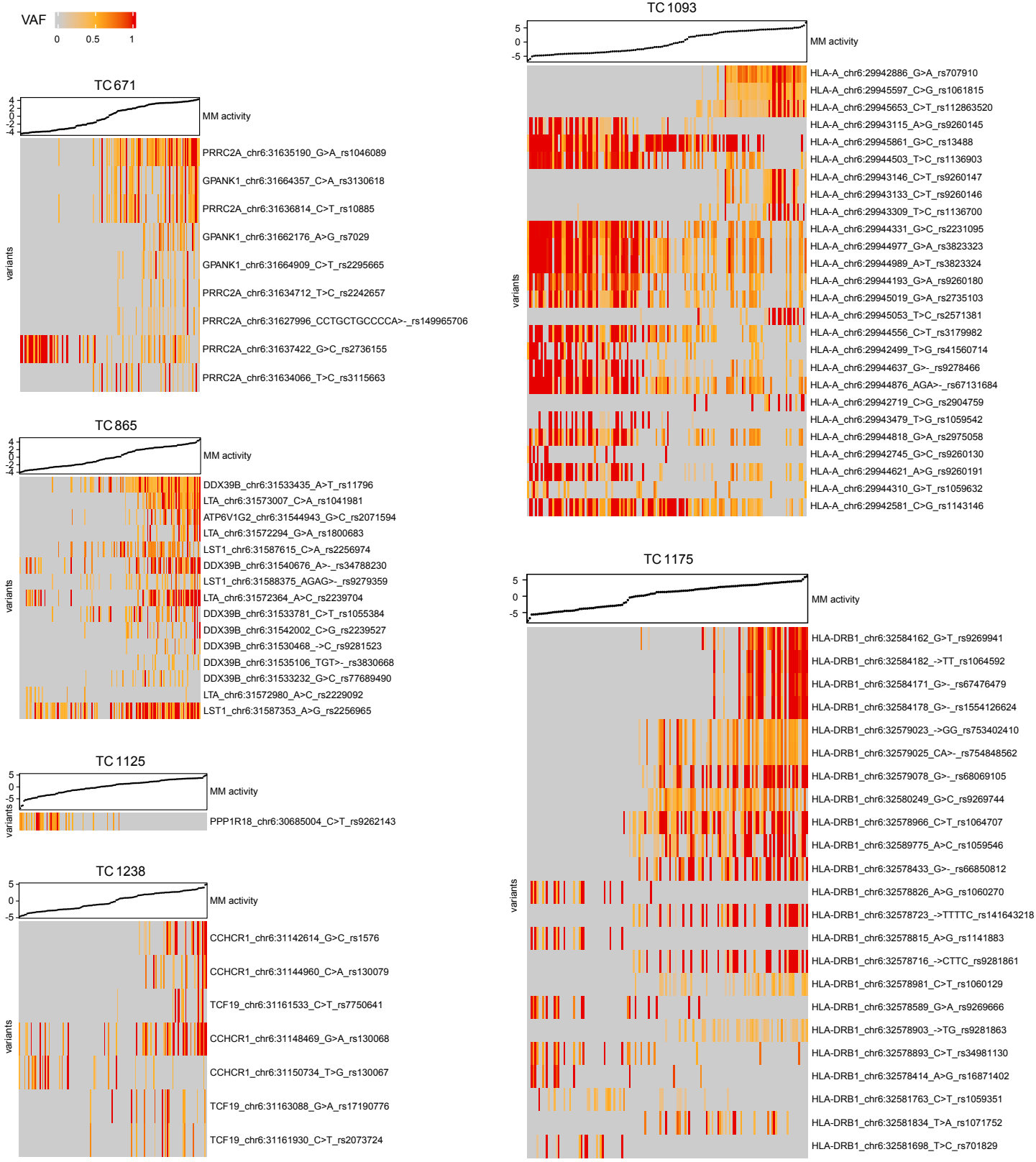


Supplementary Figure 2



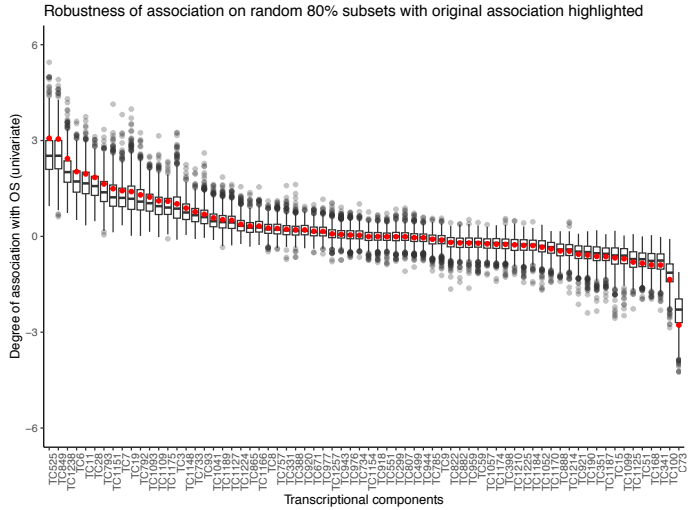
Supplementary Figure 3. Detected variants for small gene set-driven outcome-associated TCs. Columns represent samples, sorted by their mixing matrix activity for each TC. For samples with variants in the top genes of these TCs, the variant allele frequency (VAF) is shown. Rows represent variants significantly associated with mixing matrix activity, as determined by a Mann-Whitney U test with Bonferroni correction for multiple testing. Variants are ordered by decreasing statistical significance. Each row label includes gene name, chromosome, base pair position, reference base, alternative base, and reference SNP cluster ID. For TC28, no variant was significantly associated with mixing matrix activity.

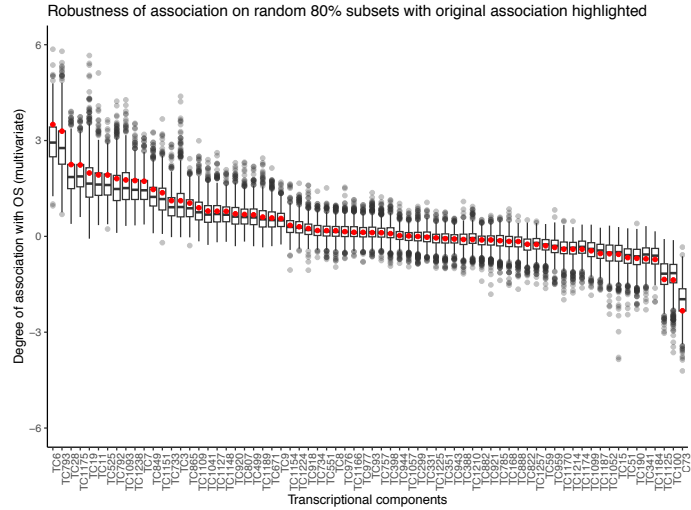
Supplementary Figure 3

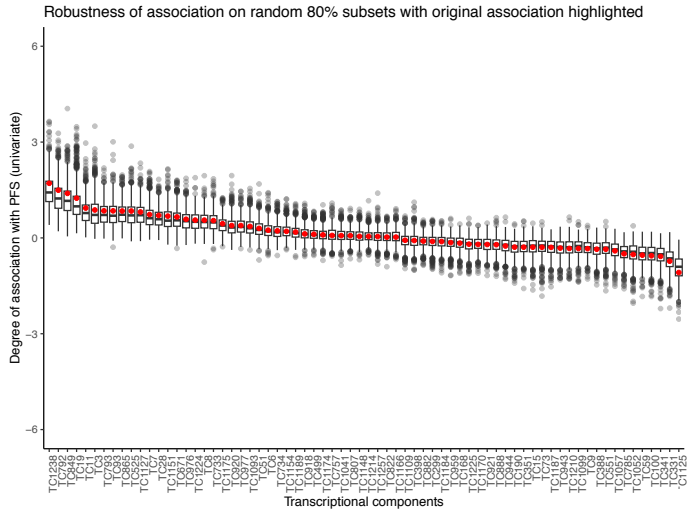


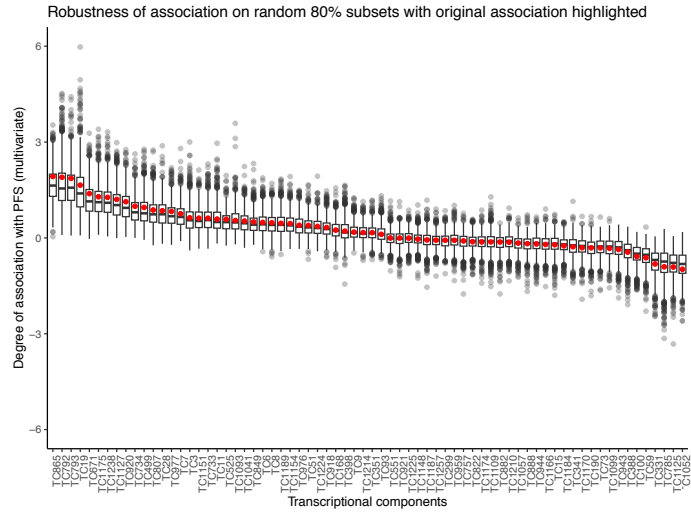
Supplementary Figure 4. Robustness of the associations between clinical outcome and TCs. The robustness of associations between TCs and outcome was determined by selecting a random 80% subsamples of the data in a permutation framework (1000 iterations). The sign and strength of associations (via coefficients and -log10 p-values) were computed for each TC. These results were compared to the original dataset's associations (red). Each plot depicts the degree of association with clinical outcome in the y-axis and the random subset of TCs in the x-axis. Each box's lower and upper ends represent the 1st and 3rd quartiles, respectively. The vertical line within the box indicates the median. Whiskers extend to 1.5 times the interquartile range below and above the 1st and 3rd quartiles. Single points represent outliers.

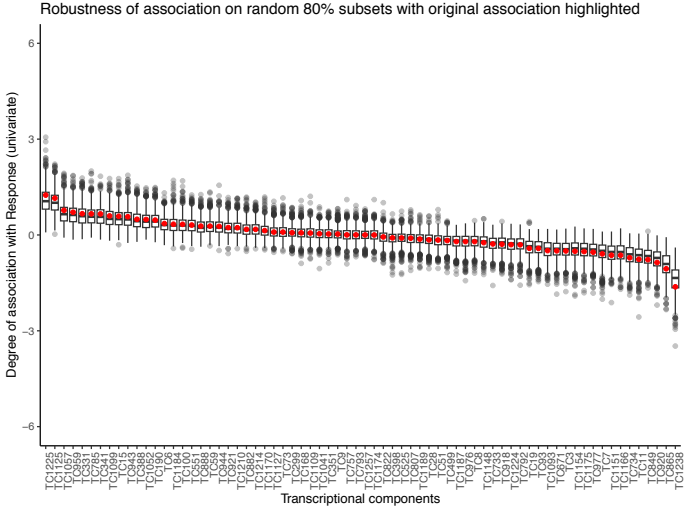
Supplementary Figure 4

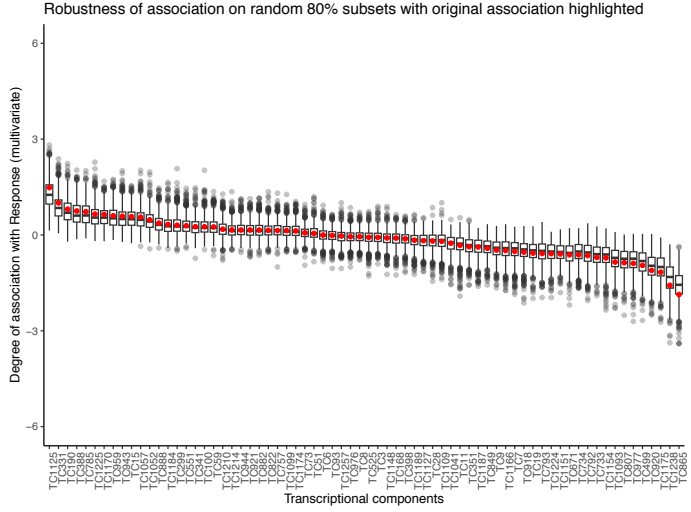






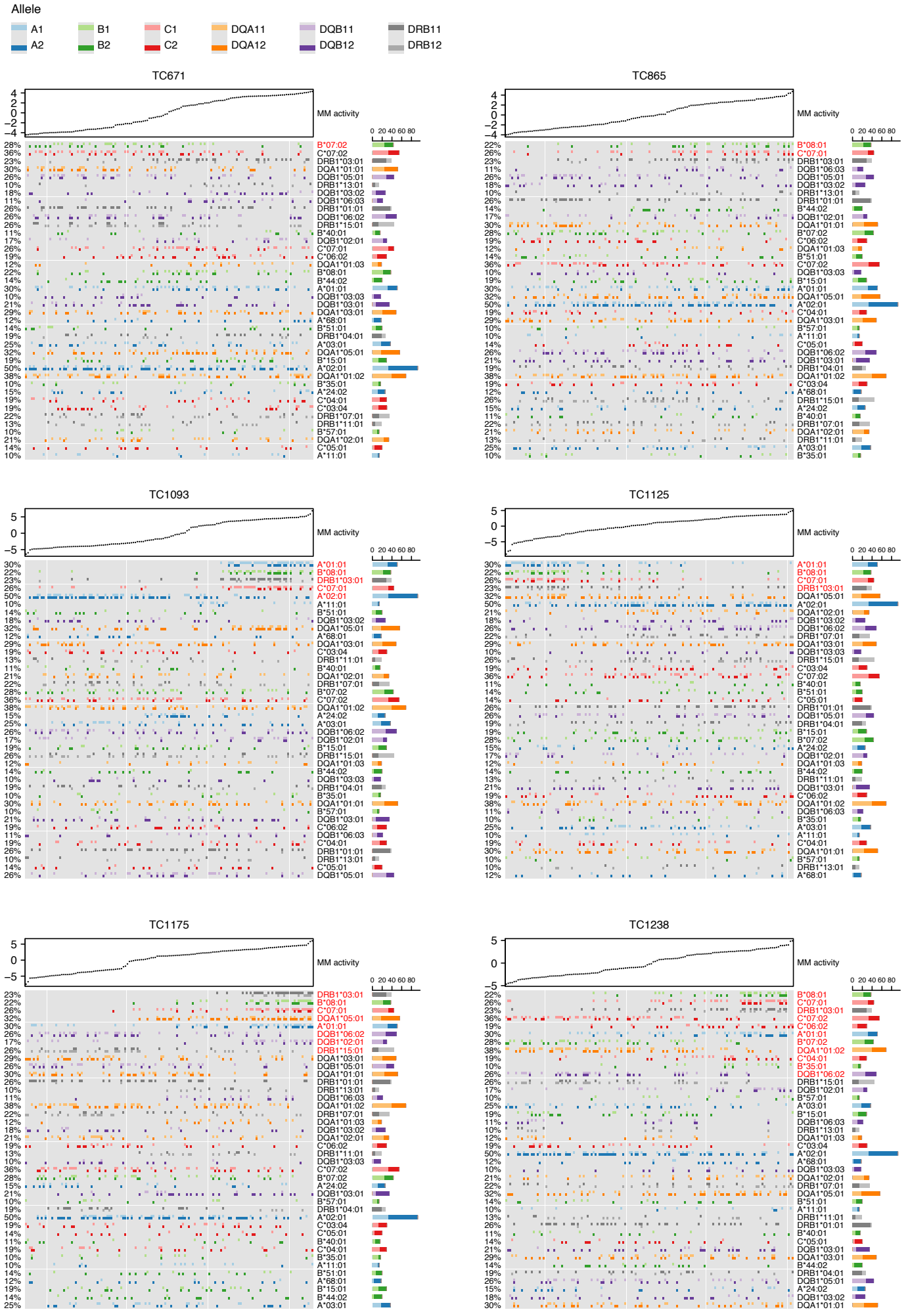






Supplementary Figure 5. HLA haplotypes for small gene set-driven outcome-associated TCs. Columns represent samples, sorted by their MM activity for each TC. Inferred HLA alleles for each sample are shown. Rows represent HLA alleles, sorted by decreasing statistical significance of their association with MM activity, as determined by a Mann-Whitney U test. HLA alleles significantly associated with MM activity and passing Bonferroni multiple testing correction are marked in red. The plots were created with an edited plotting function from the RIMA tutorial (https://liulab-dfci.github.io/RIMA/).

Supplementary Figure 5



Supplementary Figure 6. Activity of outcome-associated TCs across cell types in blood and bone marrow single-cell RNA-seq data. Single-cell RNA-seq data included 117,854 cells from 8 blood samples (**a**) and 127,618 cells from 8 bone marrow samples (**b**) from the single-cell immune cell atlas of the human hematopoietic system. The transcriptomic profile of each cell was projected onto the 18 outcome-associated TCs. Cell annotation was based on the labels defined in the immune cell atlas. Box plot colors represent major cell type groups. Each box's lower and upper ends represent the 1st and 3rd quartiles, respectively. The vertical line within the box indicates the median. Whiskers extend to 1.5 times the interquartile range below and above the 1st and 3rd quartiles. Single points represent outliers.

Supplementary Figure 6A

Supplementary Figure 6B