ABSTRACT

Title of Dissertation:	STUDIES ON VARIABILITY IN CANCER GENE EXPRESSION: FROM SINGLE PROTEINS TO POPULATIONS
	David Robert Crawford, Doctor of Philosophy, 2023
Dissertation directed by:	Stephen M. Mount, Associate Professor, Department of Cell Biology and Molecular Genetics

In this dissertation I describe four projects investigating different aspects of the variability of gene expression in human cancers. In the first chapter, we analyze epidemiological incidence rates for autoimmune diseases and cancers across numerous populations and find that sex biases in incidence rates are positively correlated between autoimmune diseases and cancers arising from the same tissue. We find that across these tissues the expression of protein-coding

mitochondrial genes is positively correlated with both autoimmune disease and cancer incidence rate sex biases, suggesting a possible direction for further investigation. In the second chapter, I construct a computational pipeline to conduct unbiased searches in large databases for possible events accounting for cancer neopeptides predicted by mass spectrometry. I identify several ribosomal frameshift-derived neopeptides from HLA-peptidomics data and discuss future approaches for further improving the accuracy and flexibility of our approach. In the third chapter, I compare the power of different multivariate Cox proportional hazards survival models based on gene- and below-gene-level expression measures to predict genes whose expression in tumor samples at diagnosis affects subsequent survival of cancer patients. I find that models based on both gene-level expression and isoform-level expression (whether transcript abundance or relative transcript abundance) identify the greatest number of statistically significant genes of interest. Finally, in the fourth chapter I briefly explore how heteroformity and entropy measures can be used to examine differences in mRNA splicing diversity at numerous levels of comparison. I propose some simple visualizations that harness these measures to display patterns in splicing diversity.

STUDIES ON VARIABILITY IN CANCER GENE EXPRESSION: FROM SINGLE PROTEINS TO POPULATIONS

by

David Robert Crawford

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor Stephen M. Mount, Co-Chair
Dr. Eytan Ruppin, Co-Chair
Professor Jonathan Dinman
Professor Philip Johnson
Professor Jiqiang Ling

Dedication

This dissertation is dedicated to my wife, Cara, who wholeheartedly supported my decision to start a new career in computational biology, and whose unwavering support ever since has made this possible.

Acknowledgements

I would like to thank my advisors Steve Mount and Eytan Ruppin for their support and guidance during my time at UMD and NCI. I would also like to thank the many members of both of their labs for their insight, collaboration, and feedback over the years.

I would like to thank my co-authors for Chapter 1 for their collaboration and insight. I would also like to thank Alejandro Schäffer for guiding me through the logistics of submitting that project for publication.

I would like to thank Yardena Samuels and Osnat Bartok for their insights and discussions on cancer immunopeptidomics research which greatly influenced my work in Chapter 2. I would (again) like to thank Alejandro Schäffer for mentoring me on how to document the design and implementation of the computational pipeline I constructed for this project.

I would like to thank Ze'ev Ronai and Sachin Verma for their experimental work testing predictions from an earlier version of the project described in Chapter 3. They and their many colleagues subsequently performed many experiments and analyses to turn one of the preliminarily promising predictions into a research paper with translational applications for cancer therapy.

I would like to thank Carl Kingsford and Hee Lee for their insights and discussions on mRNA splicing diversity during my first year at UMD which encouraged my interest in the mRNA splicing metrics discussed in Chapter 4.

Finally, I would like to thank the members of the BISI faculty and administration (especially Michelle Brooks) for getting me started in the PhD program and supporting me throughout my time at UMD.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

aeTSA	aberrantly expressed tumor-specific antigen
AID	autoimmune disease
ASB	gene set activity bias
ESB	expression sex bias
eventT	event TPM
FSu	upstream frameshift (i.e. -1 frameshift, in 5' direction)
FSd	downstream frameshift (i.e. +1 frameshift, in 3' direction)
GE	within-tissue gene expression
geneT	gene TPM
GSA	gene set activity
GSEA	gene set enrichment analysis
HLA	human leukocyte antigen
IRSB	incidence rate sex bias
isoR	isoform ratio
isoT	isoform TPM
LC-MS/MS	tandem mass spectrometry with liquid chromatography
logitPSI	logit-transformed percent spliced in
mTSA	mutation-derived tumor-specific antigen
NES	normalized enrichment score
NGS	next-generation sequencing
IQR	interquartile range
IR	incidence rate
PSI	percent spliced in
PSM	peptide-spectrum match
PSP	peptide-spectrum prediction
TPM	transcripts per million
TAA	tumor-associated antigen
TSA	tumor-specific antigen

Chapter 0: Introduction

In this dissertation I describe four projects investigating different aspects of the variability of gene expression in human cancers. Two common themes unite these projects. First, I analyze gene expression using patient- or donor-derived high-throughput sequencing data. I analyze gene expression enrichment, gene-level or below-gene-level expression, or splicing diversity based on bulk tissue RNA-seq datasets, and also gene (mis)translation based on whole proteomics and immunopeptidomics data. Second, the projects all concern topics in cancer research. I investigate gene expression associated with cancer incidence sex bias, gene translation leading to potentially immunogenic neopeptide in melanoma tumors, the use of different gene expression metrics in cancer survival models to identify potential cancer gene dependencies, or the comparison of metrics for use in comparing mRNA splicing within and between different cancers or tissues.

Large-scale next-generation sequencing projects

Next-generation sequencing (NGS) technologies have opened up the possibility of high-throughput and high-resolution DNA and RNA sequencing of biological samples [1]. In particular, high-throughput short-read RNA sequencing has made it possible to sequence tens of millions of 25- to 250-base-pair reads per sample to produce accurate, high-coverage transcriptomic datasets. In this dissertation I analyze NGS RNA-seq data in every chapter. With the exception of the small study datasets analyzed in Chapter 2, this RNA-seq data comes from two massive human tissue data projects.

The Genotype-Tissue Expression (GTEx) project was launched with the aim of providing researchers with DNA and RNA sequencing data for non-diseased human tissues [2]. The latest version (v8) published in 2017 includes data on 54 tissues from 17,382 samples from 948 donors. Tissues samples were extracted from newly deceased donors and processed for production of several data types including RNA-seq. Sample metadata includes notes on sample quality, tissue composition, and extraction site, with some pathology annotations. Subject metadata includes subject age (by decade), sex (male or female), and manner of death (using the Hardy scale). In Chapter 1 I use this data to search for tissue-specific transcriptomic patterns associated with incidence rate sex biases in autoimmune diseases and cancers. In the project described in Chapter 3 I use this data in follow-up analyses (not included in the chapter) to investigate the extent to which tumor-expressed transcripts or splicing events associated with patient survival are expressed in non-diseased tissues. In Chapter 4 I use this data to visualize mRNA splicing patterns within and between tissues.

The Cancer Genome Atlas (TCGA) project is a joint venture between the National Cancer Institute and the National Human Genome Research Institute [3]. From 2006 to 2013 the TCGA project collected paired tumor and normal tissue samples for 33 cancer types from over eleven thousand patients. Data generation was completed in 2016. Tumor samples and in some cases samples of normal adjacent tissues were collected during surgeries and processed for production of several data types including RNA-seq. Sample metadata includes notes on sample quality, tissue composition, and extraction site, amongst other values. Subject metadata is extensive --- it includes age (by decade), sex (male or female), tumor type and subtype, and extensive data on surgical history, drug treatment history, tumor remission or progression, and patient survival. In Chapter 3 I use transcript-level RNA-seq data along with patient age, sex, and

survival time to look for transcripts whose expression is associated with differential survival in metastatic melanoma patients. In Chapter 4 I use subsampled raw RNA-seq data to investigate the sensitivity of mRNA splicing diversity metrics to RNA-seq read depth. During the project reported in Chapter 2 I used raw RNA-seq data from melanoma patients to construct a melanoma-specific transcriptome to be used in constructing melanoma-specific search databases, but that part of the analysis is not discussed in this dissertation.

In the projects reported in this dissertation we use TCGA and GTEx RNA-seq data that is publicly available via the UCSC Xena project [4]. The Xena project recomputed gene expression from raw data using a single pipeline to maximize comparability between TCGA and GTEx datasets. We use GTEx RNA-seq data in Chapters 1, 3, and 4. We use TCGA RNA-seq data in Chapters 3 and 4.

Healthy tissue references for disease incidence study

In Chapter 1 we investigate connections between incidence rate sex biases in autoimmune diseases and cancers. Both autoimmune diseases (AIDs) and cancers have notably sex-biased incidence rates. We ask whether the sex biases observed in the incidence of AIDs and cancers that occur in the same tissue are correlated across human tissues. We find that the incidence rate sex biases observed for AIDs and cancers that occur in the same tissue are positively correlated across human tissues. To further investigate this correlation, we use RNA-seq data from GTEx for the twelve tissues in question. The GTEx data not only includes many samples from both men and women but it is also based on uniform data collection procedures across the tissues, increasing our confidence in cross-tissue comparisons. We analyze gene expression data from non-diseased tissue samples to determine if sex biases in gene set expression in these tissues are

correlated with AID and cancer incidence rate sex biases in the same tissues. We find that the top positively enriched gene set across human tissues whose expression sex bias is most strongly associated with the incidence rate sex biases for AIDs, cancers, and AIDs and cancers considered jointly, is the set of 37 genes encoded in the mitochondrial genome.

mRNA splicing

The increased depth and breadth of RNA-seq data from the TCGA and GTEx projects has transformed our understanding of mRNA splicing. This is possible because next-generation sequencing technologies provide enough accurate coverage of transcripts for researchers to infer not just the gene from which a transcript was transcribed but also (to a great extent) the precise structure of that transcript.

In human biology generally, alternative mRNA splicing has been shown to regulate development and tissue identity [5], and physiology and pathology [6]. However, there is continuing debate over the extent to which alternative splicing results from regulated processes [7, 8, 9] as opposed to stochastic mechanisms [10, 11, 12]. In Chapter 4 we contribute to this debate not by weighing in on one side or the other but by working to establish splicing diversity metrics that can help standardize the quantitative discussions. We adapt diversity measures from population ecology and population genetics to provide common metrics for formulating and assessing questions regarding how much splicing occurs or how much splicing changes.

It has also become clear that alternative mRNA splicing plays a big role in human cancer biology [13, 14]. RNA splicing is dysregulated in cancer [15] and this contributes to the hallmarks of cancer [16]. This opens up the possibility of targeting cancer-specific splicing events therapeutically [17], or exploiting cancer-specific splicing-derived neopeptides in anti-

cancer immunotherapy (see below) [18]. In Chapter 3 we harness transcript-level TCGA RNA-seq data to increase the predictive power of survival analyses. Survival analyses use patient data to identify potential associations between clinical characteristics and survival time. A popular approach in cancer research to identify genes of interest for further investigation is to model patient survival as a function of gene expression, age, and sex. The status quo gene expression metric for these survival analyses is gene-level RNA-seq expression level. However, some researchers have published survival analyses using the metric of isoform abundance with the aim of identifying genes of interest not found when using the metric of gene abundance [19, 20, 21, 22]. We analyze bulk RNA-seq data from TCGA metastatic melanoma samples using hierarchical multivariate Cox proportional hazards survival models to examine the predictive value of multivariate models based on different below-gene-level expression metrics in addition to subject age and sample purity. We hypothesize that some fitted models using below-gene-level metrics will identify survival associates not identified by fitted gene-level models, thus increasing the exploratory reach of gene expression-based cancer survival models.

Cancer and the immune system

Just as advances in RNA sequencing have enabled our study of mRNA splicing and expression in Chapters 3 and 4, so have advances in immunopeptidomics (also called HLA- or MHC-peptidomics) enabled our investigation into HLA-presentation of peptides derived from cancer-specific aberrant mRNA splicing events. Before going further into this data type I provide some background on immune-mediated killing of cancer cells and the various sources of abnormal peptides found in the immunopeptidome.

Immune-mediated cancer cell death

Immune reaction to tumor cells is critical for elimination of cancer. A key component of immune-mediated destruction of tumor cells is recognition by CD8+ cytotoxic T-lymphocytes (CTLs) of non-self peptides bound to major histocompatibility complex 1 (MHC-1) receptors presented on the outer membrane of tumor cells. The best-characterized of these non-self peptides or 'neopeptides', known as tumor-specific antigens (TSAs) when immunogenic, are those resulting from tumor-specific nonsynonymous mutations in the exome. Identification of specific mutation-derived TSAs (mTSAs) has led to development of mutation-specific cancer vaccines and in vitro clonal expansion and reintroduction of antigen-recognizing autologous CTLs in adoptive cell therapy [23]. In addition, the abundance of mTSAs in a tumor, the tumor mutation burden (TMB), is in many cases predictive of patient response to immune checkpoint blockade (ICB) therapy, which involves administration of antibodies that interfere with cell-mediated mechanisms of CTL response reduction [24, 25]. Here, the TMB is thought to estimate the likelihood of CTLs recognizing and attacking tumor cells once immune checkpoint inhibiting responses from those CTLs are blocked.

Immunoproteomics and neopeptides

Numerous recent studies empowered by next-generation sequencing technologies have found that various phenomena in cancers lead to aberrant RNA transcripts and (potentially antigenic) neopeptides: splicing factor dysregulation and mutation [26, 27]; increased intron retention in splicing [28]; gene fusions [29]; novel exon-exon junctions [30]; transcript escape from nonsense-mediated decay [31]; transposable element expression due to loss of DNA methylation [32]; and transcription of regions outside the exome [33, 34].

Researches have taken advantage of recent improvements in high-sensitivity mass spectrometry to profile MHC-bound peptides (collectively, the immunopeptidome) but find few if any T-cell reactive cancer-specific peptides per sample. To profile the immunopeptidome, MHC-peptide complexes are recovered from cell lysate using an immunoaffinity assay, complexes are separated and peptides eluted, and peptides are run through a mass spectrometer to produce fragmentation spectra. These spectra are then queried against a protein database using software like MaxQuant [35] to identify the peptides. For human cancer studies, the protein database is typically composed of a human reference proteome (e.g. the UniProt database) and mutated peptides predicted from whole exome or whole genome sequencing of tumor and matched normal cells. Yadav et al. [36] performed MHC-peptidomics on two mouse cancer cell lines. For the more mutated line Yadav et al. found evidence from RNA-seq for expression of 1,290 mutated peptides, 170 of which were predicted to bind to the cell line's MHC allotypes. Only 7 of these peptides were identified through mass spectrometry, and 3 were validated as immunogenic. Three 2016 studies identified mutant peptides bound to MHC molecules in human melanoma. Gloger et al. [37] isolated over 10,000 MHC-bound peptides across 5 human melanoma cell lines. They identified 262 peptides from 72 tumor-associated antigens (TAAs, immunogenic proteins overexpressed in but not exclusive to cancer), and 1 mutation-derived tumor-specific antigen (mTSA) (FDR=1%). Bassani-Sternberg et al. [38] identified 95,662 peptides across 25 human primary tumor samples, including 11 mTSAs (FDR=1%), of which 4 were found to be immunogenic by non-autologous T-cell response assays. Kalaora et al. [39] identified 4,958 MHC-bound peptides from a human melanoma sample (FDR=5%); of 5,404 peptides predicted from 1,019 nonsynonymous mutations, 2 mTSAs were identified as MHC-

bound peptides and 1 was found to be immunogenic via autologous tumor-infiltrating lymphocyte response assays.

Laumont et al. [34] carried out an expanded search for tumor-specific antigens (TSAs) in murine and human cancer cell lines by isolating MHC1-bound peptides, analyzing them via mass spectrometry, and querying a larger than normal protein database. Their proteogenomic strategy for assembling the peptide database for mass spectrometry queries attempted to capture proteins from coding as well as (normally) noncoding regions of the genome. They accomplished this by constructing their database from two sources: (1) peptides from all expressed protein-coding transcripts (determined via RNA-seq) including those bearing nonsynonymous mutations (determined via whole-exome sequencing); and (2) peptides translated from contigs of cancer-specific RNA-seq read segments. The second part differs from the first in being alignment-free --- the thinking is that since the RNA-seq reads derive from mRNA, any segment, regardless of its original genomic source (i.e. whether annotated as coding or noncoding), could potentially be translated to protein, processed, and presented by the cell's MHC receptors. They call tumor-specific antigens from noncoding regions "aberrantly expressed TSAs" (aeTSAs). Because aeTSAs need not result from mutations, it is possible this search method is more likely to find shared antigens than one focused on exom mutations. In two murine cancer cell lines Laumont et al. found 10 aeTSAs and 4 mTSAs, and 5 aeTSAs and 2 mTSAs, respectively, and they confirmed the immunogenicity of several aeTSAs *in vivo* using naive and aeTSA-vaccinated mice challenged with live murine cancer cell lines. They found aeTSAs in seven human primary tumor samples (B-ALL and lung) but did not report on their immunogenicity.

In Chapter 2 we contribute to the search for cancer neopeptides by constructing and running a pipeline for detecting aberrant peptides produced by translation perturbation. Our

pipeline constructs peptide search databases based on selected transcriptional perturbations and searches these databases for matches to *de novo* sequence predictions from HLA-peptidomics data. Here we report our search for ribosomal frameshift-derived cancer neopeptides in human metastatic melanoma tumor samples. We have also searched for neopeptides resulting from amino-acid replacements during translation but we do not report the results here. With simple modifications to the database construction procedure one can search for peptides resulting from other translation perturbation events.

Chapter 1: Sex Biases in Cancer and Autoimmune Disease Incidence Are Strongly Positively Correlated with Mitochondrial Gene Expression Across Human Tissues[1]

Abstract

Cancer occurs more frequently in men while autoimmune diseases (AIDs) occur more frequently in women. To explore whether these sex biases have a common basis, we collected 167 AID incidence studies from many countries for tissues that have both a cancer type and an AID that arise from that tissue. Analyzing a total of 182 country-specific, tissue-matched cancer-AID incidence rate sex bias data pairs, we find that, indeed, the sex biases observed in the incidence of AIDs and cancers that occur in the same tissue are positively correlated across human tissues. The common key factor whose levels across human tissues are most strongly associated with these incidence rate sex biases is the sex bias in the expression of the 37 genes encoded in the mitochondrial genome.

Introduction

Both autoimmune diseases (AIDs) and cancers have notably sex-biased incidence rates. Most AIDs occur more often in women [40, 41], and most cancers occur more often in men [42, 43, 44]. While sex differences in several key biological factors have been implicated in the biased incidence rates observed for both AIDs and cancer, including inflammation and immunity,

[1] This chapter was published in *Cancers* [174]. Any differences from the published version other than formatting changes are inadvertent.

metabolism and sex hormones, their mechanistic underpinnings remain largely unexplained [41, 45, 46].

Given these observations, we asked whether the sex biases observed in the incidence of AIDs and cancers that occur in the same tissue are correlated *across* human tissues. This question is of fundamental interest, since an affirmative answer may suggest that there are common factors underlying their incidence. Establishing such a link between AIDs and cancers could further prompt researchers to explore how pertaining findings in AIDs could cross fertilize cancer risk studies and vice versa, potentially enhancing our ability to prevent and treat these diseases.

To explore whether these sex biases are correlated across human tissues, we collected population-based AID incidence studies for tissues that have both a cancer type and an AID that arise from that tissue. For countries for which we collected AID incidence data, we gathered incidence data for corresponding cancer types from national cancer registries. Analyzing a total of 182 country-specific, tissue-matched cancer-AID incidence rate sex bias data pairs, we find that the incidence rate sex biases observed for AIDs and cancers that occur in the same tissue are positively correlated across human tissues. In addition, we analyzed gene expression data from non-diseased tissue samples to determine if sex biases in gene set expression in these tissues are correlated with AID and cancer incidence rate sex biases in the same tissues. We find that the top positively enriched gene set across human tissues whose expression sex bias is most strongly associated with the incidence rate sex biases for AIDs, cancers, and AIDs and cancers considered jointly, is the set of 37 genes encoded in the mitochondrial genome.

Methods

Overview

Our analysis is divided into two main parts: curation and analysis of disease incidence rate data; and investigation of associations between incidence rates and gene expression in corresponding non-diseased human tissue samples. First, we studied the association of incidence rates for AIDs and cancers occurring in the same tissue. We collected incidence data for AIDs from published studies, and for each country for which we found incidence data for a given AID, we collected incidence data from that country's national cancer registry for cancers occurring in the same tissue as the AID. We matched AID and cancer incidence data by tissue and by country to produce country-specific tissue-matched AID-cancer incidence rate data pairs. We then computed across-tissue correlations between AIDs and cancers for male incidence rates, female incidence rates, overall incidence rates, and incidence rate sex biases, at both the individual country level and the across-country global level.

Next, we used non-diseased human tissue transcriptomic data from GTEx version 8 [47] to investigate possible factors across human tissues that might be associated with incidence rate sex biases. We computed correlations between incidence rate sex biases and either expression of individual genes or enrichment of human functional gene sets across tissues.

Autoimmune disease incidence data curation

We first performed an extensive literature search for sex-specific incidence data for AIDs. For each AID, we searched for original studies mentioning the disease and epidemiology, prevalence, incidence, incidence rate, or sex bias using Google Scholar. We considered only population-based studies that use clinical inclusion criteria and have at least 25 cases for a given

disease. We evaluated whether or not a study was population-based using either (a) the characteristics of the existing data source used in the study (e.g., a mandatory country-wide reporting registry) or (b) estimates showing that the data collected in the study were likely representative of the overall population. We evaluated whether or not a study used clinical diagnostic criteria by looking for use of a disease-specific blood test, a histological assay, or other evidence used to confirm diagnosis and rule out similar non-autoimmune conditions. Additionally, we considered only AIDs with a focal primary tissue (e.g., we included ulcerative colitis but excluded Crohn's disease), for which we could find incidence data for at least three countries. We excluded sex-specific tissues.

We collected 188 AID-country incidence rate datasets from 167 studies. For each dataset, we calculated the *incidence rate sex bias* (*IRSB*) as

$$IRSB = log_2(IR_{MALE}/IR_{FEMALE})$$

so that a value of zero indicates no bias, a positive value indicates a higher incidence rate in males (termed a "male bias") and a negative value indicates a higher incidence rate in females (similarly termed a "female bias"). A majority of the studies provided sex-specific (123 of 188 datasets, 65%) and total (143 of 188, 76%) incidence rates (IR):

$$IR_{POP} = cases_{POP}/population_{POP}$$

where: "POP" stands for either the "MALE", "FEMALE", or "TOTAL" population; $cases_{TOTAL} = cases_{MALE} + cases_{FEMALE}$; and $population_{TOTAL} = population_{MALE} + population_{FEMALE}$). Most studies reported IR as cases per year per 10^5 persons; those using a different scale were converted to this scale. We used "crude" incidence rates (as defined above) when available; some studies provided only age-adjusted incidence rates.

Estimating incidence rates

For each of the four incidence rate measures we consider (IRSB, IR_F, IR_M, and R_{TOTAL}) the majority of studies provided a value, while other studies gave values for other measures (i.e., different incidence rates or case counts) that can be used to estimate the value of that measure. For a given measure we can divide our AID-country datasets into four groups (**Table 1**): (1) those with the measure's value but not the values of other measures we can use to estimate that value; (2) those with the measure's value and the values of other measures we can use to estimate that value; (3) those without the measure's value but with the values of other measures we can use to estimate that value; and (4) those with neither the measure's value nor the values of measures we can use to estimate that value. For each measure, we assessed the accuracy of our estimator by comparing the actual and estimated values for datasets in group (2), and then used that same estimator to estimate values for datasets in group (3).

The estimators for all four measures require a value for the population's sex ratio (Formula 1)

$$Sex_{Ratio} = \underset{(C)}{\frac{population_{FEMALE}}{population_{MALE}}} \tag{1}$$

As only one study provided the background population sex ratio, we estimated measures using either a sex ratio of 1:1 or the sex ratio for the corresponding population (matching the specific country during the years the study was conducted) according to United Nations estimates [48].

Based on (A), (B), and (C), we estimated *IRSB, IR_F, IR_M*, and *IR_{TOTAL}* as (Formulas 2−5):

$$IRSB = log_2\left(\frac{cases_{MALE}}{cases_{FEMALE}} \times Sex_{Ratio}\right) \tag{2}$$

$$IR_{FEMALE} = IR_{TOTAL} \times \frac{cases_{FEMALE}}{cases_{TOTAL}} \times (1 + 1/Sex_{Ratio}) \tag{3}$$

$$IR_{MALE} = IR_{TOTAL} \times \frac{cases_{MALE}}{cases_{TOTAL}} \times (1 + Sex_{Ratio}) \quad (4)$$

$$IR_{TOTAL} = IR_{M} \times \left(\frac{1}{1 + Sex_{Ratio}}\right) + IR_{F} \times \left(\frac{Sex_{Ratio}}{1 + Sex_{Ratio}}\right) \quad (5)$$

To assess the accuracy of our estimators we compared the actual and estimated values for datasets in group (2) in two ways (**Table 2**). First, we computed the Pearson's correlation coefficient *r* between the two values. All estimators were accurate: for each the correlation coefficient was close to 1 and the one-sided *t*-test was significant. Second, we computed a simple linear model of the form $x_{actual} = \beta \times x_{estimate} + \alpha$. All estimators were accurate: for each the coefficient β was close to 1 and the r^2 close to 1 (where *r* is the Pearson's correlation coefficient). For all four measures the estimators performed well, but for each measure the estimator using a sex ratio of 1:1 performed as good as or slightly better than the estimator using the sex ratio based on the United Nations estimates. Accordingly, for our analyses we used estimators with a sex ratio of 1:1. For all of our analyses, results computed using only given values, and not estimates, were consistent with results computed using both given and estimated values (the code for this paper includes scripts to reproduce all tests and figures using data that either includes or excludes estimated values).

Table 1. Incidence rate measures and estimators.
Measures to estimate, measures needed for estimators, and numbers of datasets with values for these measures**.** Numbers indicate dataset count and percentage (out of 188 total datasets) for each group of datasets (1–4) described in the text.

(a) Measure to Estimate	(b) Measures Needed for Estimator	(1) Datasets with (a)	(2) Datasets with (a) & (b)	(3) Datasets with (b) but not (a)	(4) Datasets with neither (a) nor (b)
IRSB	$cases_M/cases_F$	125 (66%)	105 (56%)	63 (34%)	0 (0%)
IR_M, IR_F	IR_{TOTAL}, $cases_M/cases_F$	123 (65%)	84 (45%)	41 (22%)	24 (13%)
IR_{TOTAL}	IR_M, IR_F	143 (76%)	101 (54%)	22 (12%)	23 (12%)

Table 2. Pearson's correlation coefficient and simple linear model results for estimators**.** For each measure, each of two estimators (with sex ratio as 1:1 or from the United Nations estimates) is shown with its simple linear model coefficient β, intercept α, and r^2, and its Pearson's correlation coefficient r and one-sided *t*-test *p*-value.

Measure	**Estimator**	β	α	r^2	r	p
$IRSB$	$IRSB_{1:1}$	0.922	−0.0254	0.966	0.983	6.04×10^{-78}
	$IRSB_{UN}$	0.923	−0.0585	0.963	0.981	5.54×10^{-76}
IR_{FEMALE}	$IR_{FEMALE,1:1}$	1.020	−0.303	0.997	0.998	1.15×10^{-107}
	$IR_{FEMALE,UN}$	1.035	−0.307	0.997	0.999	5.04×10^{-105}
IR_{MALE}	$IR_{MALE,1:1}$	0.975	0.228	0.997	0.999	6.10×10^{-108}
	$IR_{MALE,UN}$	0.959	0.236	0.997	0.999	2.26×10^{-105}
IR_{TOTAL}	$IR_{TOTAL;1:1}$	1.013	−0.123	1.000	1.000	1.09×10^{-182}
	$IR_{TOTAL;UN}$	1.013	−0.136	1.000	1.000	1.39×10^{-182}

Combining data for each country.

When multiple studies were available for an AID in a country, we used the across-study arithmetic mean of each incidence rate measure as the measure value for that AID-country pair (for IR_{MALE}, IR_{FEMALE}, IR_{TOTAL}, or $IRSB$ measures). Overall, surveying 167 published studies (Supplementary References), we calculated 133 country-specific AID incidence rate sex bias data points for 17 AIDs in 33 countries (Table S1, e.g., the mean incidence rate sex bias for Type 1 diabetes in Spain is one such data point).

Cancer incidence data curation

Cancer incidence rates were calculated from GLOBOCAN [49] data for all but three countries. For each country for which we had AID data, we computed each cancer type's incidence rate measure for each year and then averaged the yearly measure values to produce a single measure value for each country-cancer pair (for IR_{MALE}, IR_{FEMALE}, IR_{TOTAL}, or $IRSB$ measures). Cancer data for Finland [50], Sweden [51], and Taiwan [52] were collected from country-specific databases. For Finland and Sweden we calculated each incidence rate measure as the across-year average yearly measure for each cancer type for the most recent 20 years (1999–2019) for each country. For Taiwan we calculated each measure as the average of the measure for the two available time periods (1998–2002, 2003–2007). Overall, we calculated 165 country-specific cancer incidence rate sex bias data points for 17 cancer subtypes in 29 countries (Table S2; for an additional four countries we were unable to find population-level cancer incidence data).

Pairing AID and cancer incidence data

Across 12 human tissues we paired 17 AIDs with 17 cancer types for a total of 24 cancer-AID data pairs. To compute the correlation between AIDs and cancer incidence rate sex biases across tissues, we grouped AIDs with matched cancers occurring in the same tissue in the same country (Table S3). For example, for the UK, we paired thyroid AID data points for Hashimoto's hypothyroidism and Graves' hyperthyroidism with cancer data points for thyroid carcinoma and thyroid sarcoma, resulting in 4 possible thyroid cancer-AID pairs. The 133 country-specific AID incidence rate sex bias data points were matched to the 165 country-specific cancer incidence rate sex bias data points, yielding a total of 182 country-specific, tissue-matched cancer-AID

incidence rate sex bias data pairs that are jointly present in both the AID and cancer datasets (Table S2).

Gene expression analysis of human tissues

Gene expression was calculated from GTEx v8 data [47] provided in transcripts-per-million (TPM). For gene *i* and tissue *k* with *m* samples we calculated the within-tissue gene expression (GE) as the arithmetic mean TPM across samples as (where "POP" stands for either the "MALE", "FEMALE", or "TOTAL" population, Formulas 6−7):

$$GE_{POP,i,k} = \frac{1}{m}\sum_{j=1}^{m} TPM_{POP,i,j,k} \quad (6)$$

and the gene expression sex bias (ESB) as:

$$ESB_{i,k} = log_2\left(GE_{MALE,i,k}/GE_{FEMALE,i,k}\right) \quad (7)$$

where both $GE_{MALE,i,k}$ and $GE_{FEMALE,i,k}$ are positive.

For a set of *n* genes *N* and tissue *k* with *m* samples, we calculated the within-tissue gene set activity (GSA) as the geometric mean of gene expression across genes (Formulas 8−9):

$$GSA_{POP,N,k} = \left(\prod_{i=1}^{n} GE_{POP,i,k}\right)^{1/n} \quad (8)$$

and the gene set activity sex bias (ASB) as:

$$ASB_{N,k} = log_2\left(GSA_{MALE,N,k}/GSA_{FEMALE,N,k}\right) \quad (9)$$

where both $GSA_{MALE,N,k}$ and $GSA_{FEMALE,N,k}$ are positive.

Gene Set Enrichment Analysis across Human Functional Pathways

We performed gene set enrichment analysis (GSEA) in three steps. First, for each gene in each tissue we calculated the gene expression sex bias (ESB). Second, we computed the across-tissue Spearman correlation of the ESB of each gene with AID or cancer IRSBs (we abbreviate these correlations as $corr_{ESB/IRSB}$). We also computed aggregated or "joint" $corr_{ESB/IRSB}$ values as the average for each gene of its $corr_{ESB/IRSB}$ value for AID IRSBs and its $corr_{ESB/IRSB}$ value for cancer IRSBs. Finally, for each of these three phenotypes, we ordered all the genes from greatest to least by the $corr_{ESB/IRSB}$ values and performed a GSEA [53] to identify gene sets and pathways that were either significantly positively or negatively associated with IRSB (for gene sets used see Results). We considered GSEA results significant if the adjusted $p \leq 10^{-3}$ (we used the Benjamini-Hochberg method to adjust *p*-values for multiple tests) and ranked the results by normalized enrichment score (NES) [53].

Results

We surveyed 167 published AID studies and the cancer registries for 29 countries to assemble 182 country-specific, tissue-matched cancer-AID incidence rate sex bias data pairs (Methods; Table S2). For each study, we calculated the *incidence rate sex bias* (*IRSB*) as $IRSB = log_2(IR_{MALE}/IR_{FEMALE})$, where IR_{MALE} and IR_{FEMALE} are the male and female incidence rates, so that a value of zero indicates no bias, a positive value indicates a higher incidence rate in males (termed a "male bias") and a negative value indicates a higher incidence rate in females (similarly termed a "female bias"). Having assembled these data, we computed the mean IRSBs to get a view of tissue-matched cancer and AID incidence rate sex bias across tissues, yielding global IRSB values for 17 AIDs and 17 cancer types across 12 human tissues, comprising a total

of 24 cancer-AID data pairs. As expected, most AID incidence rates are female-biased (a negative sex-bias score), while most cancer incidence rates are male-biased (a positive sex-bias score) (**Figure 1**A, Table S4). **Figure 1**B presents the correlation of the IRSB of these disorders across human tissues, summed up across all countries surveyed. Notably, we find an overall positive correlation (Pearson correlation $r = 0.48$ with two-sided *t*-test $p = 0.017$, Spearman correlation $r = 0.43$ with two-sided *t*-test, $p = 0.034$). Repeating this analysis using various levels of cancer type classification shows a consistent and robust correlation (Figures S1 and S2, Table S5). (We used Pearson's product-moment correlation coefficient to measure correlation because it takes effect size into account. We also provide correlation test results based on Spearman's rank correlation coefficient as this assesses correlation differently and may be of interest to the reader. We considered a correlation test result significant when the *t*-test adjusted $p \leq 0.05$. We used the Benjamini-Hochberg method to adjust *p*-values for multiple tests). Second, studying this correlation in a country-specific manner for the four countries with at least 18 AID-cancer data pairs, we find a country-specific significant correlation for Sweden, while the correlations for Denmark, the UK and the USA have *q*-values (*p*-values corrected for multiple hypotheses testing) > 0.05 but are quite close to this threshold, showing a consistent trend for each country (**Figure 1**C).

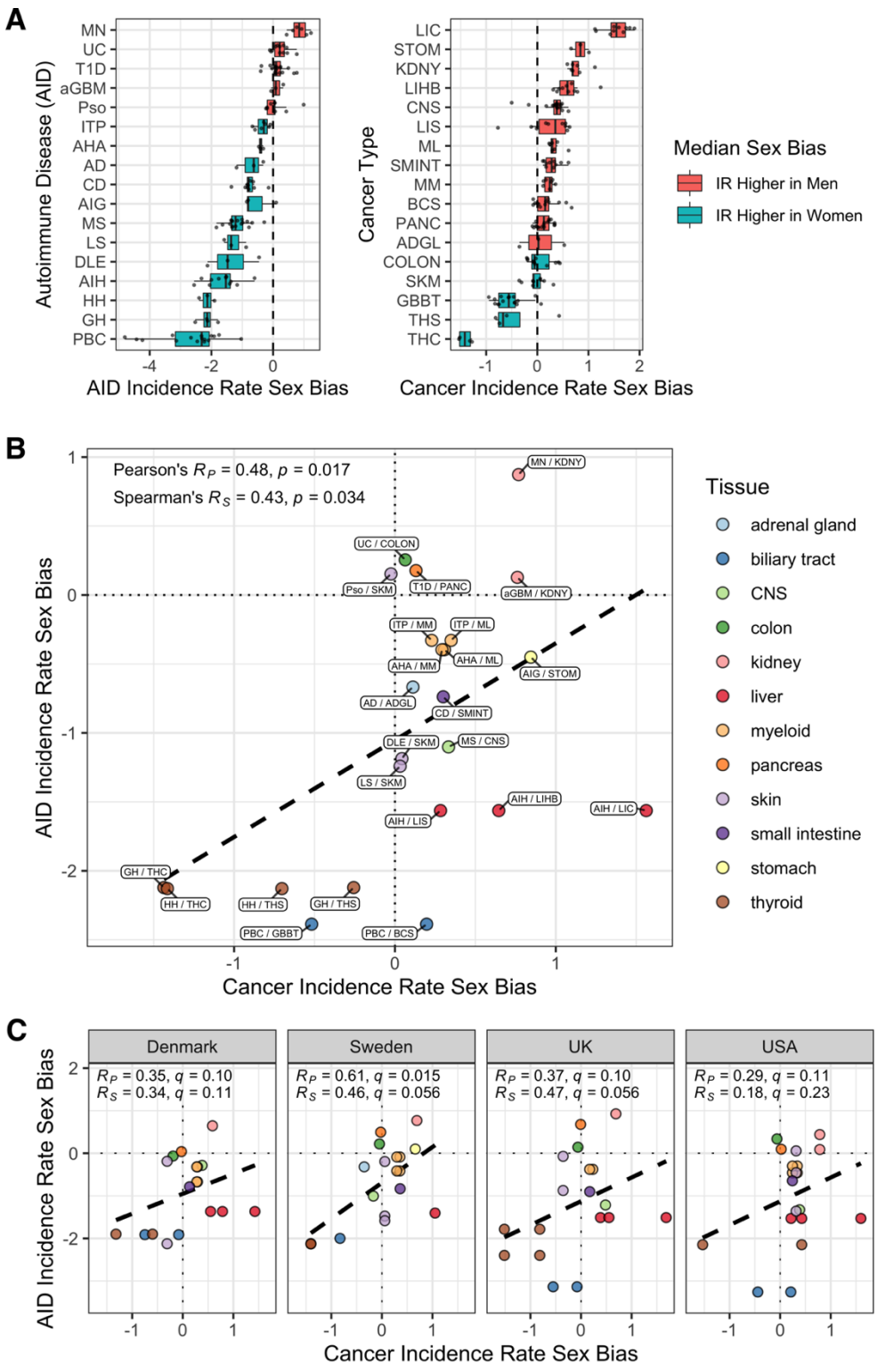


Figure 1. Incidence rate sex biases for cancers and AIDs are positively correlated across tissues of origin**.**

(**A**) Distribution across countries of incidence rate sex bias (*x*-axis) for 17 AIDs and 17 cancer types (*y*-axis). All data points are shown. Box shows interquartile range (IQR, first quartile to third quartile), with center bar representing the median (second quartile). Lefthand whisker extends from first quartile (Q1) to Q1−1.5×IQR or to the lowest value point, whichever is greater. Righthand whisker extends from third quartile (Q3) to Q3 + 1.5×IQR or to the highest value point, whichever is smaller. Positive median sex bias (red) indicates median with higher incidence rate in men; negative median sex bias (blue) indicates median with higher incidence rate in women. To fit the data compactly, the x-axes for the left and right panels differ in two unconventional ways: First, the center point for (**A**) is close to −2 because most AID IRSB values are negative whereas the center point for (**B**) is just above 0 because most cancer IRSB values are positive; Second, the range of the x-axis is greater in (**A**) than it is in (**B**) (and thus the numbering is also different) because the range of AID IRSB values is greater than that of the cancer IRSB values. (**B**,**C**) Tissue-matched incidence rate sex biases for cancers (*x*-axis) and for autoimmune diseases (*y*-axis) are displayed across different tissues of origin (circle color indicates the tissue). Positive values in each of the axes indicate male bias; negative values indicate female bias. The dashed line is the simple linear regression line. Statistics in the top left corner include the Pearson's product-moment correlation *r*-value (R_p) and *t*-test *p*-value; and the Spearman's rank correlation coefficient value (R_s) and a *t*-test *p*-value (*t*-tests were two-sided for the global-level tests and one-sided for the country-level tests). For country-level tests, *p*-values were corrected for multiple testing using the Benjamini-Hochberg method to produce q-values. (**B**) Across-population averages, with the cancer-AID pairs labeled. To fit the data compactly, the plot is centered close to (0, −1) as opposed to the more conventional center of (0,0) because the majority of the AID IRSB values are negative. (**C**) Population-level data for the four countries with the largest numbers of data pairs (at least 18 out of 24 cancer-AID pairs), maintaining the tissue color labels used in the top panel (USA, 20 pairs; Denmark, Sweden, & UK, each 18 pairs). **AIDs**: AD, Addison's disease; aGBM, anti-glomerular basement membrane nephritis; AHA, Autoimmune hemolytic anemia; AIG, Autoimmune gastritis; AIH, Autoimmune hepatitis; CD, Celiac disease; DLE, Discoid lupus erythematosus; GH, Graves' hyperthyroidism; HH, Hashimoto's hypothyroidism; ITP, Immune thrombocytopenic purpura; LS, Localized scleroderma; MN, primary autoimmune membranous nephritis; MS, Multiple sclerosis; PBC, Primary biliary cholangitis; Pso, Psoriasis; T1D, Type 1 diabetes; UC, Ulcerative colitis. **Cancers**: ADGL, adrenal gland cancer; BCS, liver (biliary) cholangiosarcoma; COLON, colon cancer; CNS, central nervous system cancer; GBBT, gallbladder & biliary tract cancer; KDNY, kidney cancer; LIC, liver carcinoma; LIHB, liver hepatoblastoma; LIS, liver sarcoma; ML, myeloid leukemia (acute and chronic); MM, multiple myeloma; PANC, pancreatic cancer; SKM, skin melanoma; SMINT, small intestine cancer; STOM, stomach cancer; THC, thyroid carcinoma; THS, thyroid sarcoma.

Observing this fundamental correlation, we next asked if we could identify factors that might jointly modulate both the incidence rate sex bias observed in cancer and in AID across human tissues. We conducted both an unbiased general investigation and a hypothesis-driven one. We specifically examined four major factors that have been previously associated in the

literature with the incidence rates of cancers and AIDs and/or their incidence rate sex biases. Those include (1) inflammatory or immune activity in the tissue [54, 55]; (2) expression of immune checkpoint genes [56, 57]; (3) the extent of X-chromosome inactivation [45, 58]; and finally, (4) mitochondrial activity [59, 60] and mitochondrial DNA copy number [61, 62].

Having these literature-driven specific hypotheses in mind, we still have chosen to begin by systematically charting the landscape of gene sets whose sex-biased enrichment in normal tissues is associated with IRSB in cancers and AIDs in an unbiased manner. We analyzed gene expression data from non-diseased tissue samples from GTEx v8 [47], for tissues in which both cancer and AID arise; GTEx data were available for 10 of the 12 tissues we studied above (Table S3). First, (1) for each gene in each tissue we calculated the *expression sex bias (ESB)* as $ESB = log_2(GE_{MALE}/GE_{FEMALE})$, where GE_{MALE} or GE_{FEMALE} denote the average gene expression in TPM (transcripts-per-million) for male or female samples of the tissue. (2) Second, we computed the correlation of the expression sex bias of each gene with AID or cancer IRSBs (we abbreviate these correlations as $corr_{ESB/IRSB}$). We also computed aggregated or "joint" $corr_{ESB/IRSB}$ values as the average for each gene of its $corr_{ESB/IRSB}$ value for AID IRSBs and its $corr_{ESB/IRSB}$ value for cancer IRSBs. (3) Finally, for each of these three phenotypes, we ranked all the genes from top to bottom by the $corr_{ESB/IRSB}$ values and performed a gene set enrichment analysis (GSEA) [53, 63] to identify gene sets and pathways that were either significantly positively or negatively associated with IRSB. In total, this analysis covered 7763 gene sets, including gene ontology biological process sets and chromosome-location based sets from MSigDB [64], three X-chromosome gene sets (fully escape X-inactivation, variably escape X-inactivation, and pseudoautosomal region) [65], and finally, the two separate sets of nuclear-encoded genes whose protein products localize to the mitochondria and the 37 mitochondrial-

genome-encoded genes [66]. In this study, we consider the 37 mitochondrial genes as a set; all the genes are listed individually and some previously identified associations between individual genes and either AIDs or cancer are shown in Table S6.

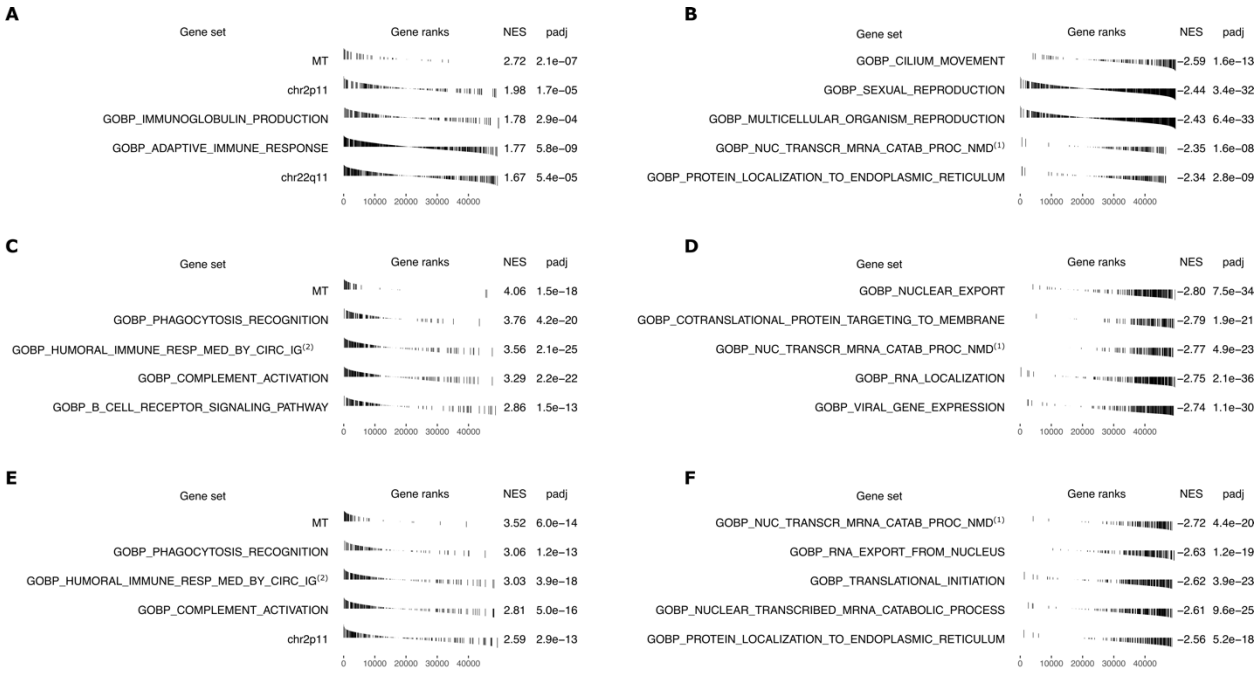


Figure 2. GSEA results for correlations of gene expression sex bias with IRSB across 10 GTEx tissues.
Top 5 positively and negatively enriched gene sets, with adjusted $p \leq 10^{-3}$, for AID incidence (positive, (**A**); negative, (**B**)), cancer incidence (positive, (**C**); negative, (**D**), and AIDs and cancers jointly (positive, (**E**); negative, (**F**)). For each gene set the plot shows: (Gene set) name; (Gene ranks) bar plot of $\text{corr}_{\text{ESB/IRSB}}$ values ordered from highest correlation at the left to lowest at the right (bars for genes in the gene set are black); (NES) normalized enrichment score; and (padj) Benjamini-Hochberg corrected p-value. Abbreviated gene set names: (1) Gene Ontology Biological Process (GOBP) Nuclear transcribed mRNA catabolic process nonsense-mediated decay; (2) GOBP Humoral immune response mediated by circulating immunoglobulin; (3) GOBP Nuclear transcribed mRNA catabolic process. Tissues include: adrenal gland, brain, colon, kidney, liver, pancreas, skin, small intestine, stomach, and thyroid.

Figure 2 shows the top positively and negatively $corr_{ESB/IRSB}$ enriched sets with $p \leq 10^{-3}$ after multiple hypotheses test correction for AID incidence (positive, Panel A; negative

Panel B), for cancer incidence (positive, Panel C; negative Panel D), and their joint aggregate enrichment for both AID and cancer incidence (positive, Panel E, negative, Panel F). Strikingly, the top enriched gene set (highest normalized enrichment score (NES)) in *all* three phenotypes is the set of 37 genes encoded on the mitochondrial genome, including many genes with high $corr_{ESB/IRSB}$ values. In contrast, while the (much larger) set of all genes encoding proteins that localize to the mitochondria is significantly enriched for cancer IRSB, it is not significantly enriched for AID IRSB, where it is only ranked 3842 out of 6420 (negatively) enriched gene sets. Several immune-related gene sets also show high and significant $corr_{ESB/IRSB}$ positive enrichments in accordance with one of our initial hypotheses (**Figure 2**). However, the three different X chromosome gene sets studied in light of another one of our original hypotheses are not significantly enriched in $corr_{ESB/IRSB}$ values. Finally, several mRNA processing gene sets show strong negative significant correlations and high negative NES scores with AID and cancer incidence.

To obtain a clearer visualization of the key positively enriched gene sets described above, we summarized the expression of the genes composing a given gene set in a normal GTEx tissue by computing their geometric mean, giving us a single activity summary value. We then computed the correlation across tissues between these summary values of the gene sets in each normal tissue and the IRSBs of cancers or AIDs (**Figure 3**). In concordance with the results of the unbiased analysis presented above, we do not observe a significant correlation between cancer or AID incidence rate sex bias and the expression of key immune checkpoint genes (CTLA-4, PD-1, or PD-L1, Figure S3), or the extent of X-chromosome inactivation (quantified by the expression of XIST lncRNA [67], Figure S4). We also do not find such significant consistent correlations for the top immune gene sets found via the unbiased analysis (previously

shown in Figure 2). However, we do find strong correlations between these summary values for the mitochondrial gene set, which was ranked highest in Figure 2 (gene set "MT"): Remarkably, we find that the sex bias of mtRNA expression in GTEx tissues is positively correlated both with AID incidence rate sex bias (Pearson $r = 0.56$, one-sided t-test $p = 0.018$) and with cancer incidence rate sex bias (Pearson r = 0.67, one-sided t-test $p = 0.0058$) (**Figure 3**A,B; the correlations between mtRNA expression and cancer and AID incidence rates for each of the sexes individually are provided in Figure S5). The significance of these two associations is further supported by observing that the basic correlation between cancer and AID IRSBs becomes insignificant when we compute the partial correlation between these two variables while controlling for the mtRNA expression bias (Pearson r = 0.21, two-sided t-test $p = 0.42$). Overall, these findings are in line with previous reports linking mitochondrial activity [59, 60] and mtDNA copy number [62, 61] with higher AID and cancer incidence.

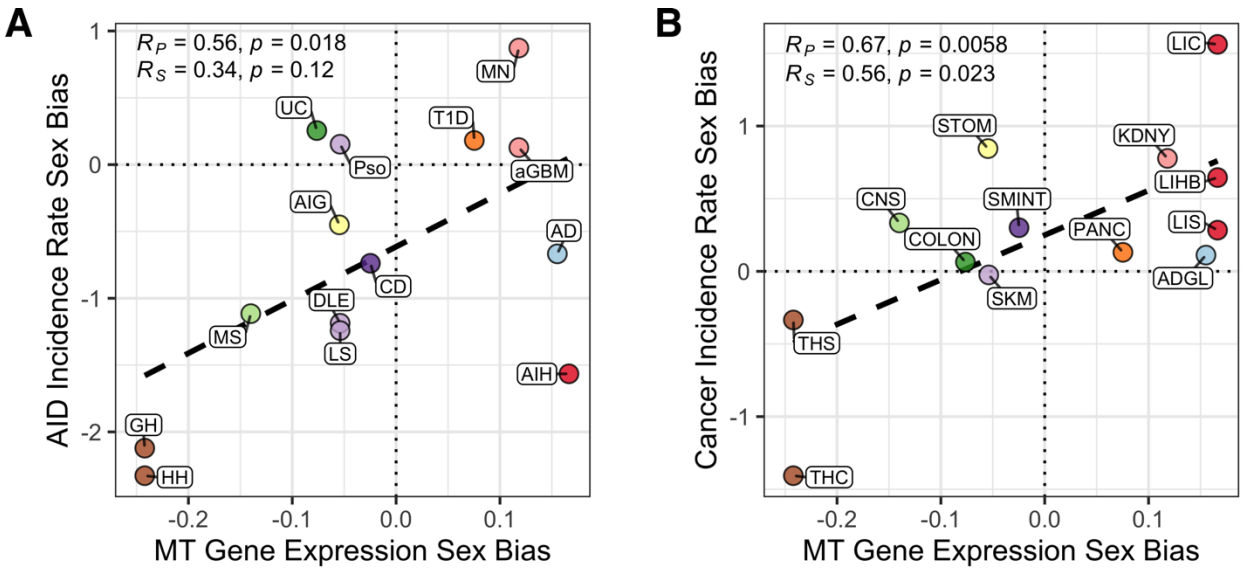


Figure 3. Mitochondrial gene expression is a strong correlate of sex biases in incidence rate of autoimmune diseases and cancer types across tissues.

The correlation between expression ratio of mitochondrial gene expression in male vs. female tissues (*x*-axis) with the incidence rate sex biases of (**A**) autoimmune diseases (*y*-axis) and (**B**) cancer types (*y*-axis) across human tissues (circle color indicates the tissue). AIDs: AD, Addison's disease; aGBM, anti-glomerular basement membrane nephritis; AIG, Autoimmune gastritis; AIH, Autoimmune hepatitis; CD, Celiac disease; DLE, Discoid lupus erythematosus; GH, Graves' hyperthyroidism; HH, Hashimoto's hypothyroidism; LS, Localized scleroderma; MN, primary autoimmune membranous nephritis; MS, Multiple sclerosis; Pso, Psoriasis; T1D, Type 1 diabetes; UC, Ulcerative colitis. Cancers: ADGL, adrenal gland cancer; CNS, central nervous system cancer; COLON, colon cancer; KDNY, kidney cancer; LIC, liver carcinoma; LIHB, liver hepatoblastoma; LIS, liver sarcoma; PANC, pancreatic cancer; SMINT, small intestine cancer; SKM, skin melanoma; STOM, stomach cancer; THC, thyroid carcinoma; THS, thyroid sarcoma.

Discussion

The correlative findings between the expression of mitochondrially encoded genes and cancer and AID IRSBs across human tissues are quite surprising, giving rise to two further fundamental questions. First, what biological mechanisms may be associated with sex differences in overall mitochondrial functioning? One potential candidate may be estrogen signaling, which has been shown to regulate at least four mitochondrial functions relevant to health and disease [68], including, (1) biogenesis of mitochondria, whose levels differ across sexes and tissues [69], (2) T-cell metabolism (including mitochondrial activity measured by Seahorse assays) and T-cell survival (estimated by retention of inner membrane potential) [70], (3) unfolded protein response [71] (mediated partly via mitochondrial superoxide dismutase) [72], and (4) generation of reactive oxygen species (ROS) [73]. Second, how might sex differences in mitochondria functioning modulate the sex-biased incidence observed in cancers and AIDs? One possible mechanism is through differences in ROS production, which notably involves quite a few mitochondrially encoded genes: Increased mitochondrial ROS generation has been associated with both the initiation and intensification of autoimmunity in several organ-specific AIDs [59]

and with cancer initiation and progression [60]. More generally, alterations in mtDNA copy number have been associated with increased risk of lymphoma and breast cancer [62], and somatic mtDNA mutations producing mutated peptides may trigger autoimmunity [61].

Our analyses have a few limitations and we list three main ones. First, the majority of our AID-cancer data pairs are from European countries (113 of 182 [62%]), which might introduce geographic, ethnic, or social biases. This geographic bias in our AID-cancer data pairs is largely due to a paucity of suitable AID epidemiological studies based on populations outside of North America and Europe, particularly on populations in low- and middle-income countries [74, 75]. For example, despite the geographically widespread study of common AIDs such as multiple sclerosis and type 1 diabetes, for other common AIDs such as Hashimoto's hypothyroidism, Graves' hyperthyroidism, and ulcerative colitis, data from regions outside North America and Europe are sparse [76]. Second, factors beyond biological drivers, such as sex differences in the propensity to seek medical care or reporting of specific diseases, are not characterized in the datasets studied. However, putative disease-specific effects may be somewhat mitigated given the opposite tendency of sex biases for AIDs and cancers in a study of tissue-specific correlations like ours. Although there is evidence for the role of environmental exposures in the development of some AIDs [77], there is little evidence for sex-specific exposures contributing to sex biases in AID incidence [78]. Likewise, a recent study of the contribution of risk factors to sex disparities in the incidence of solid tumor cancers at 21 anatomical sites found that differences between male and female incidence rates are largely unexplained by factors outside of sex-related biological factors [79]. Third, although much of the incidence rate data is age-standardized, we could not take additional steps to account for age-related incidence rate

differences as the sample sizes available are too small to enable doing such an analysis in a robust manner.

It has been hypothesized that chronic inflammation leads to cancer [80, 81]. Indeed, a recent study analyzing UK Biobank data found positive associations between several tissue-specific immune-mediated diseases (i.e., diseases such as asthma and myositis in addition to AIDs) and subsequent cancer risk in the same individual [82]; they did not however consider sex-bias. Both this study and our study seek statistical evidence of shared risks between immune diseases and cancer. However, in contrast to this individual-level study design assessing within-subject risk of sequential disease appearance in the UK population, we chose a population-level study design assessing within-population correlations between IRSBs for AIDs and cancers affecting the same tissues across populations. A strength of our study is that when AID studies reported whether individuals developing AIDs had previous immune-related diseases or cancers we excluded individuals with such previous diseases from our study.

As in humans, sex differences have been reported in animal studies of diseases, which has prompted us to search the literature and survey previous studies of sex bias in disease incidence in rodent models of cancers and AIDs. We focused on studies of sex difference in spontaneous and/or autochthonous carcinogenesis by either carcinogen treatment or genetic engineering, excluding transplantation of syngeneic animals because these animals do not model disease development (representative examples are listed in Tables S7 and S8 for cancer and AIDs, respectively). Table S7 lists our cancer incidence findings, where the sex bias was male skewed in colon, liver, kidney, pancreas, and stomach, and higher in females in the thyroid, consistent with the human reports. Interestingly, for colon, liver, kidney, pancreas, and thyroid, the sex bias disappeared or was reduced when the animals were subjected to

castration/ovariectomy or hormone treatment, supporting the notion that the differences in these organs are likely to be driven by sex hormones. Table S8 lists AID rodent models that allow for direct comparisons to the human data. The AID sex bias reported is however generally higher in males than in females, in difference from the human findings, but the higher male bias observed in kidney, colon, pancreas, and skin compared to the thyroid is maintained.

Conclusion

In summary, we find a surprising overall positive correlation between cancer and AID incidence rate sex biases across many different human tissues. Among key factors that have been previously associated with sex bias in either AID or cancer incidence, we find that the sex bias in the expression of mitochondrially encoded genes (and possibly in the expression of a few immune pathways) stands out as a key factor whose aggregate level across human tissues is quite strongly associated with these incidence rate sex biases. Our findings thus call for further mechanistic studies on the role of mitochondrial gene expression in determining cancer and AID incidence and their incidence rate sex biases.

Publication backmatter

Supplementary Materials

The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/cancers14235885/s1, Figure S1: The correlation between incidence rate sex bias for cancers and AIDs (using low-resolution cancer classification); Figure S2: The correlation between incidence rate sex biases for cancer types and AIDs (using high-resolution cancer classification); Figure S3: The correlation between checkpoint gene expression

sex bias versus AID and cancer incidence rate sex bias; Figure S4: The correlation between XIST gene expression versus AID and cancer incidence rate sex bias; Figure S5: The correlation between mitochondrial gene expression versus AID and cancer incidence rates in males and females; Table S1: Curated autoimmune disease study data; Table S2: Tissue-based cancer and autoimmune disease pairings; Table S3: Matched cancer and autoimmune disease data pairs; Table S4: Cancer and AID incidence rate averages; Table S5: Expanded cancer classifications; Table S6: The 37 genes encoded on the human mitochondrial genome, and some known associations between individual genes and either autoimmune diseases or cancers; Table S7: Curated cancer rodent model data; Table S8: Curated autoimmune disease rodent model data. Supplementary references. References and DOI links for studies curated in Table S1.

Author Contributions

Conceptualization, D.R.C. and E.R.; methodology, D.R.C., S.S., B.M.R., J.S.B.- S., P.E.C., P.S.R., S.M.M., A.E., K.A., A.A.S. and E.R.; software, D.R.C. and S.S.; validation, D.R.C. and S.S.; formal analysis, D.R.C., S.S. and N.U.N.; investigation, D.R.C., P.S.R., C.-P.D. and A.A.S.; data curation, D.R.C. and C.-P.D.; writing—original draft preparation, D.R.C. and E.R.; writing—review and editing, D.R.C., S.S., N.U.N., B.M.R., J.S.B.-S., S.M.M., A.E., K.A., P.E.C., P.S.R., C.-P.D., A.A.S. and E.R.; visualization, D.R.C.; supervision, A.A.S., S.M.M. and E.R. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

All data used is publicly available. Code and data (including links to original sources, raw data downloaded from those sources, and processed data files) used for analysis are available on

Zenodo at DOI: https://doi.org/10.5281/zenodo.7058954 (last accessed on 16 September 2022). Statistical analyses and figure preparation were performed on a Macintosh computer (OS 12.5.1; 32 GB memory; 8-core 2.3 GHz processor) in RStudio (v2021.09.0 + 351 "Ghost Orchid" release) [83] running the R language (v4.1.2) [84] (R package versions used are listed in the Zenodo repository). Plots produced in R were aligned and lettered using Inkscape (v1.0) (inkscape.org (last accessed on 16 September 2022)) to produce multi-plot figures.

Acknowledgments

This research was supported in part by the Intramural Research Program of the NIH, NCI. This work utilized the computational resources of the NIH HPC Biowulf cluster (http://hpc.nih.gov (last accessed on 16 September 2022)).

Chapter 2: Combining *De Novo* Peptide Sequencing Predictions with Massive Database Searches to Identify Cancer Neopeptides

Introduction

The evolutionarily conserved systems for translating mRNAs into proteins are often perturbed in cancer cells [85]. This can provide selective advantages to cancer cells, but by producing aberrant protein that can be presented on a cancer cell's HLA receptors it can also trigger an anti-cancer immune response. Two recent studies demonstrated that both interferon-gamma (IFNg) treatment and tryptophan (W) depletion can cause translation perturbation in melanoma cells [86, 87]. Immune cytolytic T-cell activation leads to release of IFNg which induces melanoma cells to catabolize tryptophan and export the resulting immunosuppressive catabolites [86, 88]. This immunosuppressive response by the cancer cell causes a shortage of the already rare tryptophan and can result in ribosomal stalling at tryptophan codons during translation, leading to ribosomal frameshifts [86] or incorporation of a different amino acid (with a near-cognate anticodon) into the growing peptide [87]. Both processes can produce aberrant peptides capable of stimulating an anti-cancer immune response.

In this chapter we describe our computational pipeline for detecting aberrant peptides produced by translation perturbation. We use published RNA-seq and mass spectrometry (MS) data from Bartok et al.'s study of MD55A3 melanoma cell lines that underwent either IFNg treatment or W-depletion [86]. We construct large databases of possible ribosomal frameshift-induced aberrant peptides and search for matching sequences in mass spectrometry data from HLA-bound peptides. Although this computational approach to finding these rare events in proteomics data is quite challenging [89], we aim to demonstrate the feasibility of this approach

and point towards future work that can make this a practical method for identifying novel forms of translation perturbation in cancer.

Frameshift events

Frameshift events occur when a ribosome adjusts its position upstream (towards the 5' end of the transcript) or downstream (towards the 3' end of the transcript) on the mRNA transcript, thereby changing its codon reference frame. The most common frameshifts are "-1" frameshifts (a shift by one nucleotide upstream; here we call these "FSu" events) and "+1" frameshifts (a shift by one nucleotide downstream; here we call these "FSd" events). In the best studied cases frameshift events result from multiple evolutionarily conserved mRNA features that together cause ribosomes to stall and then shift upstream (or less commonly downstream) [90, 89, 91]. In contrast to this "programmed" frameshifting, the less-studied process amino acid shortage-induced frameshifting appears to rely less on mRNA sequence features other than occurrence of specific codons and its occurrence with respect to particular transcripts more haphazard [91]. In either case however we formally represent the products of these events in the same manner. In **Figure 4** (a) we see an RNA transcript (here we use the nucleotide thymine, 'T', in the place of uracil, 'U') divided into three-nucleotide codons according to each of the three possible reading frames. The protein products in (b) result from translation without a frameshift. In (c) we represent a +1 ribosomal frameshift after the underlined codon 'GTT' by removing the first nucleotide of the next codon ('T') because in this case the ribosome moves forward by one nucleotide and so never reads that nucleotide, instead first reading the codon 'GTT' in Frame 1 and then reading the codon 'GAC' in Frame 2. In (d) we represent a -1 ribosomal frameshift after the underlined codon 'GTT' by doubling the last nucleotide of that codon ('T') because in this

case the ribosome moves backwards and so reads the last nucleotide of that codon twice, first as the last letter of 'GTT' in Frame 1 and second as the first letter of 'TTG' in Frame 0.

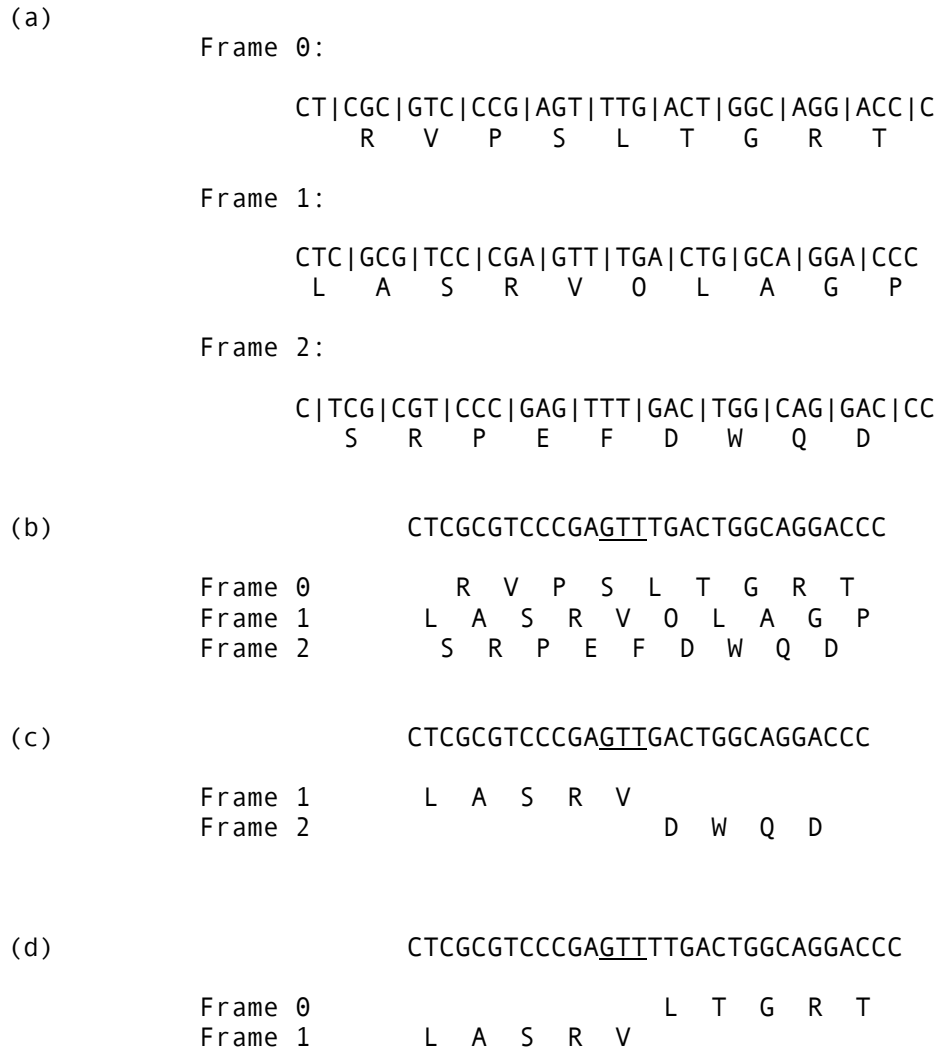


Figure 4. Reading frames and frameshifts.
We represent the translation of the stop codon 'TGA' with 'O' (see Methods).

Experimental data

Bartok et al. studied the human melanoma cell line MD55A3 *in vitro*. Four replicates were made for each of three conditions: (1) no treatment (NT); (2) IFNg treatment (IFN); and (3)

tryptophan-depleted medium (W) (going forward we will refer to each of these replicates as an 'experiment'). For each of these twelve experiments they extracted HLA-bound peptides. In addition they performed RNA-seq for two replicates each of the NT and IFN conditions.

Mass spectrometry

Bartok et al. analyzed the HLA-bound peptides isolated from each experiment using liquid chromatography with tandem mass spectrometry (LC-MS/MS). The output of the LC-MS/MS is the starting point of our pipeline. Liquid chromatography involves elution of the peptides for two hours in a linear gradient to stratify them based mostly on molecular polarity. As peptides reach the end of the gradient (this time point marks their "retention time") they are ionized and sprayed into the first mass spectrometer (MS1) which separates them according to their mass-to-charge ratio (m/z) (see

Figure 5 top panel). In our case, the first spectrometer selects high-intensity portions of each m/z window ("data-dependent acquisition") to send for fragmentation. These "precursor" peptides are fragmented by high-energy collision-induced dissociation (HCD) so that amino acid bonds are broken (typically one bond is broken per individual peptide). The fragments are sent to the second mass spectrometer (MS2) where the fragments are separated according to mass-to-charge ratio (m/z) (see

Figure 5 middle panel). In the end, we have the LC retention time and mass (and m/z) for the precursor peptides (from MS1) and the mass (and m/z) for the fragment peptides (from MS2) for each precursor. (Unlike with many protein mass spectrometry analyses, because we start with short peptides (<20 amino acids) we do not require the use of enzymatic digestion to break up larger proteins into suitably small peptides.)

We can predict peptide sequences from LC-MS/MS data using established MS analysis tools. Sequencing involves assigning a peptide sequence to a MS scan (a peptide-spectrum prediction, or PSP) that satisfies several constraints: (1) the peptide mass is close to the precursor mass from MS1 (within some error tolerance window); (2) the fragments from MS2 are consistent with the precursor peptide sequence and show adequate coverage of possible fragments (there is also noise to deal with), especially the predicted most likely fragments; and (3) predicted amino acid masses may include allowed post-translational modifications or fragmentation modifications. LC-MS/MS cannot distinguish between isoleucine ('I') and leucine ('L') amino acids because they have the same mass. Accordingly, peptide sequence prediction tools typically predict 'L' for the ambiguous (iso)leucine positions with the understanding that downstream analysis may be used to disambiguate those positions.

Peptide sequence prediction tools employ one of two approaches. First, tools like MaxQuant [35, 92] start with the MS data and a reference protein database and compute which peptides from the database are likely to produce MS scans similar to the experimental scans. For each scan these tools yield a ranked list of peptide-spectrum matches (PSMs), where scoring is based on a confidence p-value and the proportion of the predicted peptide's theoretically expected MS2 fragments that were actually found in the MS2 scan. Second, tools like PepNovo [93] start with the MS data and for each scan find the best peptide matches out of all possible peptide sequences (any sequence assembled from the twenty amino acids). For each scan these tools yield a list of peptide-spectrum predictions (PSPs) ranked according to the overall confidence of the prediction (a function of p-value), which itself is derived from (typically the arithmetic mean of) the confidence values for the individual amino acid predictions.

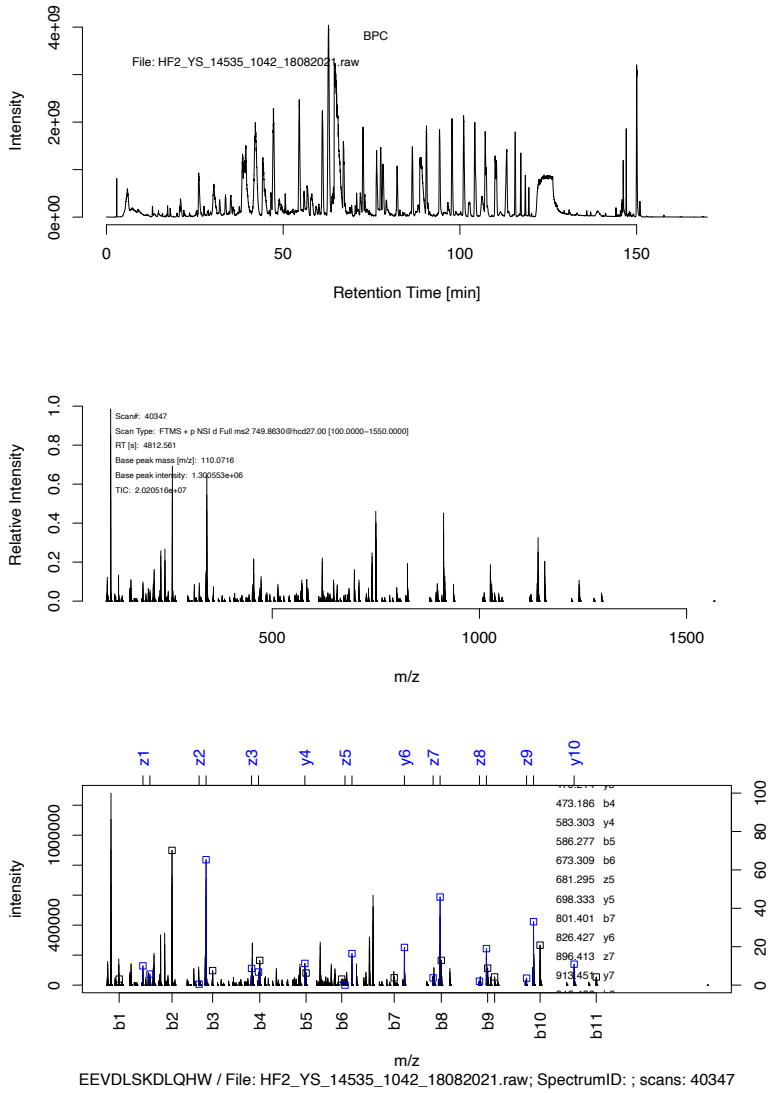


Figure 5. LC-MS/MS output examples.
(Top) The base peak chromatogram for a MS run shows the retention time and intensity for precursor ions analyzed by MS1. (Middle) This plot shows the relative intensity and *m/z* from MS2 scan of fragments of precursor ions with a retention time of 80.21 and *m/z* of 110.07. (Bottom) This plot shows the predicted fragments for a peptide predicted for this spectrum, 'EEVDLSKDLQHW', overlayed on the MS2 scan plot. (Top and middle plots produced using *R* package 'rawrr' [94]; bottom plot produced using *R* package 'protViz' [95].)

We opt for the second sequence prediction approach because our search space is far too large to be accommodated by the database approach (the set of possible peptides including amino

acid replacements or frameshifts is orders of magnitude larger than the set of possible peptides in reference proteomes). We divide the task into two steps. First, we use *de novo* tools to derive PSPs from our MS data. Second, we search for these predicted peptides in a custom database to yield the subset of these which are PSMs. This presents us with three challenges: (1) we must construct the large custom database; (2) we must efficiently search the custom database; and (3) we must measure and control false-discovery rate in this large search space.

False matches are a concern, especially when we search for a peptide in a very large database so that random matches are likely. Accordingly we employ a strategy to estimate our rate of false matches at any given peptide score. Our goal is to estimate for peptides matched at or above a given score the percentage of those matches that are false positives. This percentage is the false discovery rate (FDR) at that score and characterizes not any individual match but rather the collection of matches at or above that score. In our case we employ a more stringent FDR Q score, which estimates for a given peptide score the maximum FDR for it or any greater peptide score (in this way the FDR Q score is monotonic whereas the local FDR can increase or decrease depending on the specific peptide score windows compared).

To compute our false-discovery rate (FDR) we use the widely employed target-decoy competition (TDC) method [96]. We construct a decoy database of peptides from our reference or 'target' database and search for peptide-spectrum matches in the combined target-decoy database. We assume the distribution of scores for PSMs to the target database is a mixture of scores for false and for true identifications. Under the assumptions that no peptides are found in both the target and decoy databases (the "no overlap" condition) and that false identifications are equally likely in both target and decoy databases (the "equal probability" condition), we estimate

the distribution of false identifications in the target database as the distribution of false identifications in the decoy database.

Methods

Design

Outline

The pipeline takes as input raw mass spectrometry data (and optionally RNA-seq data) from an experiment and to identify differentially expressed proteomic and/or MHC-bound neopeptides derived from frameshift or amino acid replacement events. (MHC is the more general term of which HLA is the human-specific form. Since we are dealing only with human cells in this project, we use the two terms interchangeably.) The pipeline includes eight modules, whose dependencies are shown in **Figure 6**. In Module 1, we compute peptide-spectrum predictions using *de novo* MS sequencing tools applied to our raw MS data. In Module 2, we process the predicted peptides from Module 1 to produce the peptide sequences used in downstream modules. In Module 3, we construct peptide databases based on: three reference human proteomes; cell line-specific somatic mutation calls; common MS contaminant proteins; and proteins derived from *in silico* three-frame translation, with and without frameshift events, of a reference human transcriptome. In Module 4, we search for the query peptides in our peptide databases to produce peptide-spectrum matches. In Module 5, we use an MHC binding strength prediction tool to predict the binding strengths of our query peptides to the cell line specific MHC allotypes. In Module 6, we use the outputs of Modules 4 and 5 to filter our peptide-spectrum matches and produce an FDR-controlled list of identified peptides. In Module 7, (optional) we quantify gene expression in each experiment using raw RNA-seq data. Finally, in

Module 8 we combine the outputs of Modules 4 through 7 to test our hypotheses regarding the impact of experimental conditions on frameshift events (or in another application, amino acid replacement events).

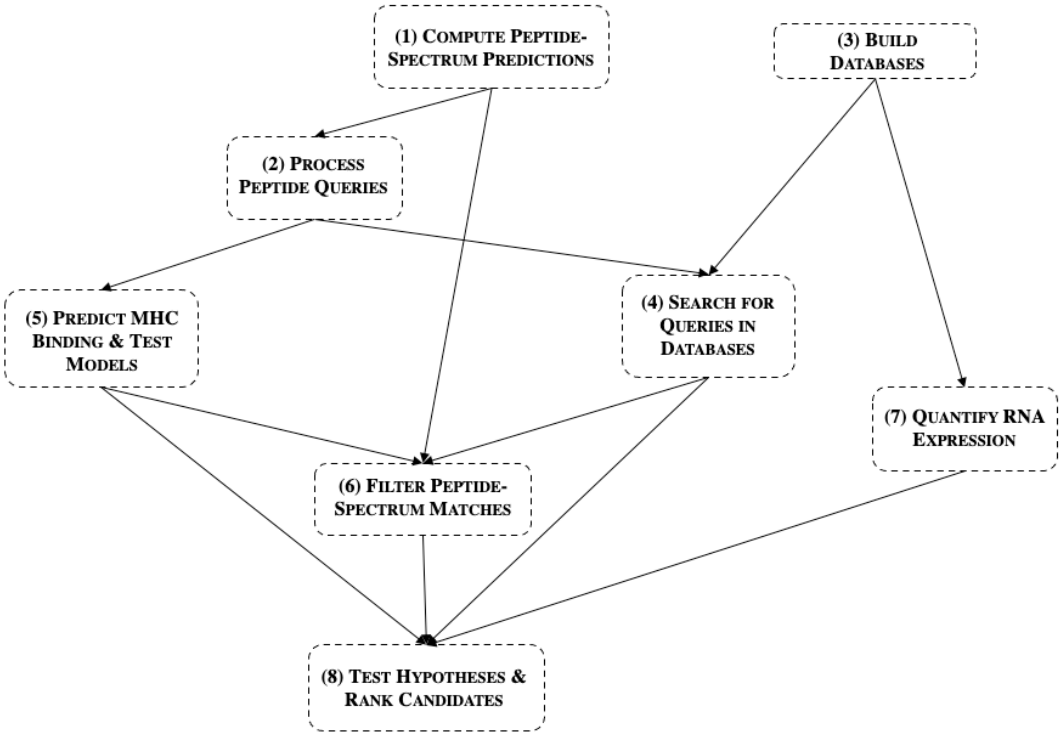


Figure 6. Simple dependency graph for Snakemake modules.

1. Compute Peptide-Spectrum Predictions

We use the open-source *Casanovo* de novo prediction tool to compute peptide-spectrum predictions from our raw MS data [97, 98]. This tool predicts a single peptide for each MS2 spectrum and assigns each amino acid in this predicted peptide a confidence score (ranging from 0 to 0.99). We compute the peptide-level average of this confidence score and use this peptide-level score to rank this peptide-spectrum prediction (PSP).

2. Process Peptide Queries

In this module we take as input the peptide-spectrum predictions from module 1 and prepare the predicted peptides for searching against our peptide databases. We process the input peptides ("Peptide_MS") in three ways (

Table 3): First, we remove post-translational-modification prediction notation to yield a simple amino-acid-only peptide sequence for each peptide ("Peptide_clean"). Second, we compute the possible isoleucine-leucine variants for each peptide ("Peptide_IL"). Third, we compute the reverse of each peptide sequence to form the decoy query set ("Peptide_TD"). Once we have the target and decoy peptide sets, we separate the queries by length and sort these length-specific sets. The Peptide_MS sequence predicted by our MS analysis tool can include leucine (L) but not isoleucine (I) residues as mentioned earlier. Accordingly, for each Peptide_simple we compute each possible Peptide_IL that results from one or more L residues in Peptide_simple being replaced by an I residue. For a sequence with *n* leucines we will add have 2^n-1 Peptide_IL variants (not counting the original). Thus for each Peptide_simple there is always at least one Peptide_IL sequence that is equivalent to Peptide_simple, but there may be others that are not. The set of Peptide_ILs represents all possible actual peptides from the experiment. We use this set for retention time modeling, MHC-binding prediction, and searching our protein databases (whether canonical or novel). When we search for Peptide_IL queries in the peptide databases, a match for any Peptide_IL derived from a Peptide_simple is a match for that Peptide_simple.

Table 3. Example peptide forms.

	Peptide_MS	Peptide_clean	Peptide_IL	Peptide_TD
(a)	YTDATSKY	YTDATSKY	YTDATSKY	YTDATSKY
	"	"	"	YKSTADTY
(b)	S(+42.01)VDYKKKY	SVDYKKKY	SVDYKKKY	SVDYKKKY
	"	"	"	YKKKYDVS
(c)	SEFEKNKKL	SEFEKNKKL	SEFEKNKKL	SEFEKNKKL
	"	"	"	LKKNKEFES
	"	"	SEFEKNKKI	SEFEKNKKI
	"	"	"	IKKNKEFES

3. Build Databases

We construct peptide databases from human proteome references and a human transcriptome reference. To include as many reference proteins as possible in our reference proteome database [99] we start from the intersection of three publicly available human proteome references: RefSeq, UnitProt, and Gencode. We also use a common protein contaminants file available from the MaxQuant website. We use transcript sequences to compute the following protein sequences for each of three translation frames: (1) the translation product of the entire transcript; (2) the translation product for each possible +1 frameshift event; and (3) the translation product for each possible -1 frameshift event. (In principle one could include any translation event perturbation at this step.)

We adopt a data-partitioning parallelism approach [100] to make our database processing and searching more efficient. We partition each proteome or peptidome into subsets so that (a) the union of those subsets is the entire proteome and (b) any given protein is found in exactly one

partition (the partitioning method is irrelevant so long as those two conditions hold). We then process each proteome partition into sets of unique fixed-length peptides for lengths 8-14 (**Figure 7**).

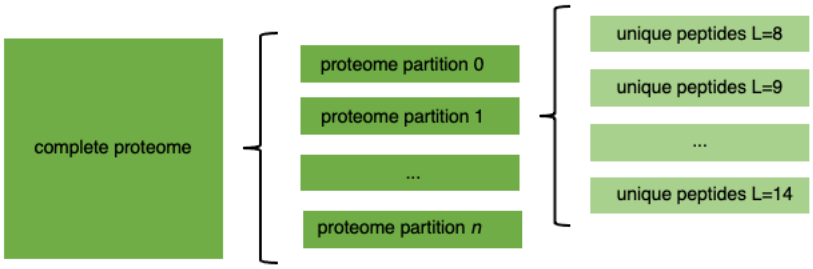


Figure 7. Database partition structure.
We built our databases using peptide lengths 8 through 14 but only analyzed peptides of length 9 (by far the most common in HLA peptidomics outputs) for this chapter.

4. Search for Queries in Databases

We employ a two-step search to efficiently search for the queries in the large protein databases. Each database is divided into partitions and each is searched separately. The first step is the membership search, which tells us whether a given query peptide is anywhere in the protein partition. This step is fast. The output is a list of query peptides. The second step is the location search, which uses a text search to find every location of each query match in the protein partition. Compared to the first step, for each peptide this search is relatively slow. In practice, with a large database containing tons of rare events (like our frameshift databases) the majority of queries are dropped (not found) in the membership search. In our pipeline, we search using the intersection of all queries from all files (for each letter pair) for efficiency, and then analyze the individual experiments one by one.

We use the Bash join function to compute the intersection of the query peptide set and the database partition peptide set for a given length. Peptides in the intersection are found somewhere in the database partition, but determining their precise location(s) in the database partition proteins requires the location search. Peptides found by the membership search to be in the database partition are located precisely using a string search. The function searches for each query peptide throughout the entire database partition, returning protein IDs which provide the peptide source: gene, transcript, event type, and event location. The output from this step is used only in the final result ranking step.

5. Predict MHC Binding

We use the neural network-based MHC binding prediction tool NetMHC-4.0 [101]. This tool tools take as input the MHC allotypes of the samples (the cell lines) and the list of peptide queries and for each predicts the binding strength of each peptide to each MHC allotype (in nM IC50, also given as an allotype-specific percentile rank, where lower is stronger).

6. Filter Peptide-Spectrum Matches

Here we compute metrics used to filter our PSMs. For each peptide, we compare its predicted retention time to its actual retention time as reported in the LC-MS/MS output. We predict retention time using the SSRC function [95] based on the Kyte & Doolittle protein hydropathy scale [102]. We build a linear model of the form pred_RT ~ RT+constant and identify those peptides whose residual under this model is an outlier. We also compute the Shannon-Weaver entropy for each peptide sequence for later use.

7. Quantify RNA expression

We compute gene- and transcript-level expression based on raw RNA-seq data. This information can be used to filter results in Module 8.

8. Rank candidates and test hypotheses

Here we compare results across all of our samples and test our specific hypotheses. We compute individual sample-database search FDR Q scores using different filter sets. This step is done locally using easily customized scripts that run quickly.

Implementation

System setup

The pipeline is set to run on the NIH Biowulf HPC cluster (a linux environment), but can be adapted to other cluster setups if we change the snakemake syntax for submitting cluster jobs and the command line syntax for loading outside utility modules.

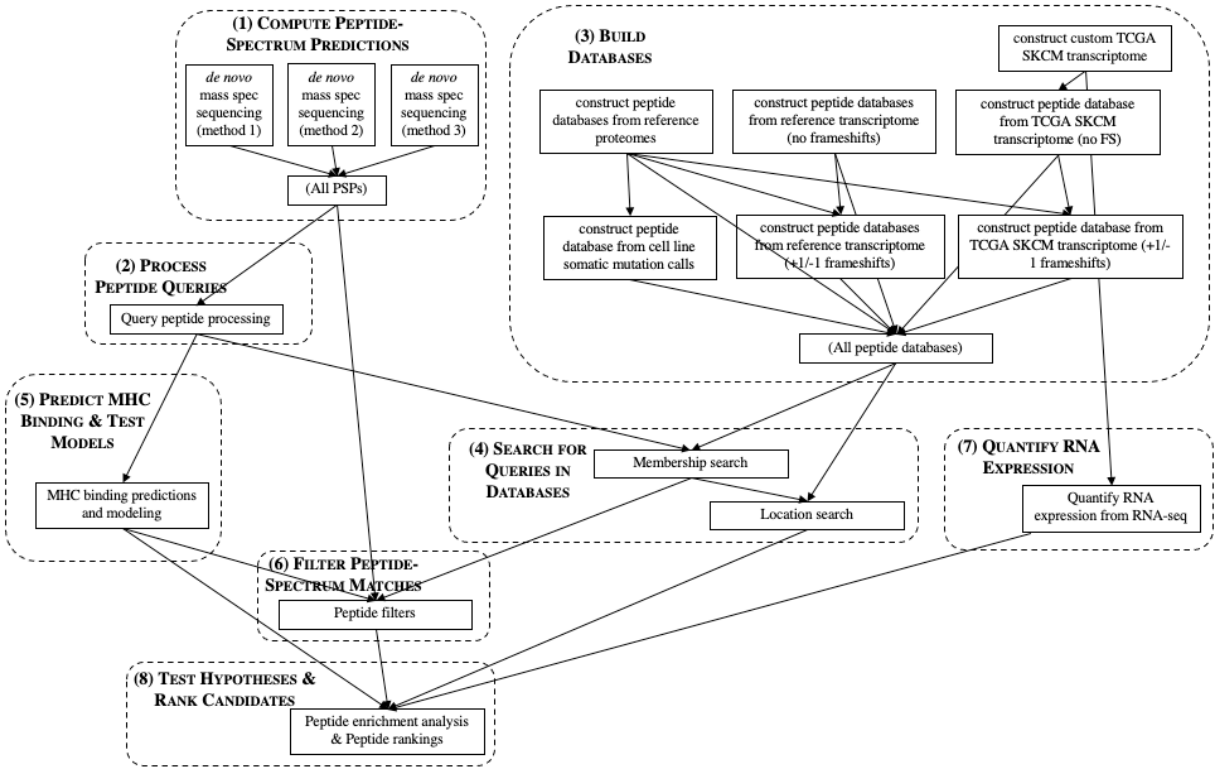


Figure 8. Detailed dependency graph for Snakemake modules.

We performed the custom TCGA SKCM transcriptome steps in Module 3 but do not discuss them in this chapter and did not further analyze those outputs. We performed *de novo* MS sequencing in Module 1 using two methods but only discuss the results for the *Casanovo* tool in this chapter.

Workflow management

Each of the major steps in the pipeline is carried out using an individual Snakemake script [103]. Snakemake is a workflow management system that: (1) produces a directed acyclic graph from a set of processing steps based on their inputs and outputs to arrive at an appropriate order of implementation; (2) creates directories needed for declared output files; (3) manages cluster job submission and checks for output files; and (4) runs in Bash strict mode so that any errors thrown during implementation of a rule cause that run to fail (this prevents some errors from slipping past). **Figure 8** shows a directed acyclic graph representing the dependencies of module

components on one another. If a given module requires as input the output of another module, then that other module is shown with an arrow pointing to the given (dependent) module.

Typesetting code

In this section we typeset computer code in `Andale Mono` and indent it from the left. The code language is given in square brackets in the regular text preceding the indented code (languages include *Bash*, *Awk*, *R*, and *C++*). A line of code that is meant to be entered on the command line shell is preceded by ">>"; code without this precedent is either *R* or *C++* code excerpted from a larger code file. For command line code we use square brackets to denote an argument that needs to be specified, e.g. `[input.tbl]`. Square brackets can mean something else in shell but we do not use them at all in our command line code so there is no ambiguity in the code and command lines that follow.

Coding languages

Snakemake scripts [103] are written in a Python-based language (the Snakemake language is a superset of Python). We do not include Snakemake code in this discussion of the pipeline. We wrote custom utilities in C++ following the C++11 standard and compiled them on the HPC cluster using the following command [*Bash*]:

```
>> g++ --std=c++11 -o [utility] [utility.cpp]
```

We use *Bash* and *Awk* in the command line. We use the following *Bash* commands: sort, join, uniq, cat, wc, shuf, and ls. We use simple *Awk* instructions for text manipulation and two *Awk* scripts for file conversions. We use *Rscript* from *R* version 4.1 to run *R* scripts from the command line.

Hardware

We designed the pipeline to run on the NIH Biowulf High Performance Computing cluster. We designed the compute-intensive modules (all except modules 6 and 8) to efficiently use Biowulf resources by either: combining numerous small, independent tasks into larger groups for job submission; or partitioning larger jobs into numerous, smaller, independent tasks for job submission. The former approach avoids overtaxing the Biowulf job allocation system; the latter approach is a form of data partitioning parallelism [100] and allows us to submit jobs to the 'quick' partition, which allocates smaller unused portions of compute nodes for relatively short jobs.

1. Compute Peptide-Spectrum Predictions

We download MS raw data files from the PRIDE EBI database using *pridepy* (https://github.com/PRIDE-Archive/pridepy), a python client for the PRIDE Archive Rest API. To retrieve the raw data files deposited by Bartok et al. (under ID PXD020224) we run [*Bash*]:

```
>>  module load python/3.7
>> pridepy download-all-raw-files -a PXD020224 -o
/data/crawforddar/transfid_snakemake/PRIDE_EBI_files/PXD020224/
```

To remove the 64-character unique file ID (and a trailing "-" for a total of 65 characters) from PRIDE files (the rest of the file name is sufficient in this case) we run the following from the raw data file download directory [*Bash*]:

```
for f in *; do mv "$f" "${f:65}"; done
```

We use ThermoRawFileParser [104], an open source tool that uses an API from Thermo Scientific to convert their proprietary RAW format to an open format. This gives us metadata

and the human-readable open format, like MGF, that we can use as input for the *de novo* sequencing tool. We run [*Bash*]:

```
>> module load mono/5.18.1
>> mono trfp/ThermoRawFileParser.exe -i {input} -b {output} -f 0 \\
      -m 1 -c {output_meta}
```

We download and install Casanovo [97, 98] and built its Conda environment. To test Casanovo we run a Biowulf interactive session with a k80 GPU node, allocating 10G of local scratch, 20G of memory, and 14 CPUs:

```
>> source /data/$USER/conda/etc/profile.d/conda.sh
>> source /data/$USER/conda/etc/profile.d/mamba.sh
>> conda activate casanovo_env
>> sinteractive --gres=gpu:k80:1,lscratch:10 --mem=20g -c14
```

We run the following command [*Bash*]:

```
>> casanovo --mode=denovo --model=casanovo_massivekb.ckpt \\
      --peak_path=sample_data/sample_preprocessed_spectra.mgf
```

We use these same system resources and the same commands (except for the "sinteractive" allocation) to process all of our raw MS files via SLURM jobs submitted through our Snakemake framework.

2. Process Peptide Queries

We extract from each Casanovo output file the following for each PSP: MS1 and MS2 scan index numbers; predicted peptide sequence; predicted peptide mass (can deviate slightly from the mass reported in the raw MS file); and the confidence score for each amino acid in the predicted sequence. To process the predicted peptide sequence we first use a regular expression in *R* to remove any post-translational modification marks from the sequence to produce Peptide_simple consisting solely of amino acid letters [*R*]:

```
psp_dt[,Peptide_simple:=str_replace_all(Peptide_MS, "[^A-Z]", "")]
```

For example, in

Table 3 row (b) the substring "(+42.01)" representing a predicted 42 Dalton acetylation is removed, whereas for rows (a) and (c) no changes are made and Peptide_simple is equivalent to Peptide_MS.

Next, we remove spectra that do not have at least one predicted peptide within our length range of interest, 8 to 14 amino acids [*R*]:

```
spectrum_vec = unique(peaks_dt[Peptide_length>7 & Peptide_length<15,
      Spec_id])
psp_slim = psp_dt[Spec_id %in% spectrum_vec,]
```

For use in later steps, we determine the range of lengths of all predictions for these spectra [*R*]:

```
kmin = min(psp_slim$Peptide_length)
kmax = max(psp_slim$Peptide_length)
```

Finally, we take the unique set of the simple peptides for further processing [*R*]:

```
pep_uniq = list(unique(psp_slim$Peptide_simple))
```

We pass the unique list of Peptide_simple as intermediate input to process using our *C++* tools to produce isoleucine-leucine variants (Peptide_IL). For each Peptide_simple we compute the list of all possible I/L variants, Peptide_IL. Since isoleucine (I) and leucine (L) have the same mass, they are indistinguishable in the MS/MS analysis, and thus Casanovo simply assigns "L" for this ambiguous position, with the understanding that any 'L' could actually be an 'I' in the source peptide. To compute all possible L-to-I changes we use a recursive function in a *C++* utility, called with [*Bash*]:

```
>> build_IL_variants [input_file] [output_file]
```

The recursive function called for each sequence is as follows [*C++*]:

```
int recursive_IL_swap(ofstream & the_stream, ofstream & the_dict,
      string original_line, string the_line, int the_position){

      auto line_length = the_line.size();
      int new_position=the_position + 1;

      if(new_position > line_length){

            the_stream << the_line << "\n";
            the_dict << original_line << "\t" << the_line << "\n";

      } else{

            string current_letter = the_line.substr(the_position, 1);

            if( current_letter == "L" ) {

                  string header = the_line.substr(0, the_position);
                  string tail = the_line.substr(new_position,
                        line_length-new_position);
                  string sequence_sub = header + "I" + tail;
                  recursive_IL_swap(the_stream, the_dict,
                        original_line, sequence_sub, new_position);

            }

            recursive_IL_swap(the_stream, the_dict, original_line,
                  the_line, new_position);
      }
      return(0);
}
```

Next, we reverse each of our target sequences to produce a decoy search set to complement our target set (together, Peptide_TD) for use in our target-decoy competition analysis [96] using a *C++* utility [*Bash*]:

```
>> build_reverse_tbl [target.tbl] [decoy.tbl]
```

We construct a reverse query set and search for it in our target databases instead of constructing a reverse (decoy) database and searching for our target queries in the decoy database because the end results are the same but the former is more computationally efficient (with respect to both building and storing the decoy set). We use a custom *C++* utility instead of a simple *R* string reversal function because running the *R* function for millions of sequences is inefficient.

We partition the query set by length using a *C++* utility [*Bash*]:

```
>> partition_kmers_simple [query_seqs.tbl] [kmax] [kmin] [out_dir]
```

The output files are single-column files listing peptide sequences. We sort each of the single-length files (sorting is required for later use as input to the *Bash* join function) [*Bash*]:

```
>> sort [unsorted_peptides.tbl] > [sorted_peptides.tbl]
```

3. Build Databases

To construct our *in silico* translated databases for three-frame translations without frameshifts ("noFS" database), with +1 frameshifts ("FSd" database), and with -1 frameshifts ("FSu" database) we use the Gencode v43 transcriptome based on genome assembly version GRCh38.p13, downloaded from the Gencode website [105]. We first partition the transcriptome into sets using the *C++* utility "build_nt_parts_graded". This utility first divides the transcriptome into files by transcript length and then divides each length-specific file into subsets of a certain size to facilitate our data-partitioning parallelism for our database construction and database search modules. The transcript lengths and partition sizes (**Table 4**) were selected based on several rounds of trial-and-error to ensure that individual jobs would run quickly enough to be allocated to the 'quick' partition on Biowulf.

Table 4. Transcriptome partition parameters

Transcript length (nt)	**Maximum transcripts per partition**
0-5000	5,000
5001-10,000	250
10,001-15,000	25
15,001-20,000	5
20,001+	2

The partitioning script is run as [*Bash*]:

```
>> build_nt_parts_graded [module_dir] [sample_name]
```

This is run from within the Snakemake module and uses the sample_name and module_dir to build paths for the input file, the intermediate files, and the output files.

To translate each transcriptome partition into three separate protein sets we call the *C++* utility "build_prot_X_for_stop" using the command line [*Bash*]:

```
>> build_prot_X_for_stop [input_file] [output_tag] [partition_id_num]
[kmax]
```

where 'input file' is a two-column tbl file with transcript label and transcript (nucleotide) sequence, 'output_tag' is the protein database name (e.g. "Gencode_v43"), and 'partition_id_num' is the number of the transcriptome partition used. The script produces three protein.tbl files, one with proteins for each of three reading frames and without frameshift events; one with one protein for each of three reading frames and for each possible +1 frameshift event; and one with one protein for each of three reading frames and for each possible -1 frameshift event.

We perform *in silico* transcript translation with a *C++* utility script that separates the transcripts into three-nucleotide codons and then for each codon either appends to the translated peptide sequence either the standard 1-letter code for its amino acid (we use the IUPAC standard one-letter amino acid code [106]; e.g. 'C' stands for Cysteine and is coded by 'TGT' and 'TGC'). We translate stop codons using three letters not assigned in the IUPAC one-letter code: 'J' for 'TAA', 'O' for 'TGA', and 'U' for 'TAG'. We translate stop codons as letters for two reasons. First, if one is interested in readthrough due to nonsense suppression by near-cognate tRNAs [107] or increased expression of certain tRNAs [108] one can write a script to go through the translation files and change 'J', 'O', or 'U' to the common amino acid replacements for these stop codons.

Second, it allows us to construct a single protein sequence for each nucleotide sequence while still marking stop sites. In module 6 ("Filter Peptide-Spectrum Matches") we can use location search results from module 4 to locate the first stop or start codon upstream of a given peptide (to see if its expression would require stop codon readthrough).

We produce a list of chimeric peptides for each fixed length *L* using a *C++* utility [*Bash*]:

```
>> build_kmers_window [input.tbl] [output.tbl] [L]
```

For each translated peptide, the utility trims the sequence on either side of the frameshift event (shown in

Table 5 as a vertical line) for each peptide to length *L*-1 so that each subpeptide of length *L* extracted from this peptide contains at least one amino acid on the lefthand and on the righthand side sof the frameshift event. A peptide trimmed in this manner has a length of at most 2(*L*-1) and includes at most *L*-1 distinct subpeptides of length *L*.

Table 5. Extracting chimeric frameshift peptides

```
(a)   ...DEILIRDTFRTYCQERLMPRIL|LANRNEVFHREIISEMGE...

(b)             DTFRTYCQERLMPRIL|LANRNEVFHREIISEM

(c)             DTFRTYCQERLMPRIL|L
                 TFRTYCQERLMPRIL|LA
                  FRTYCQERLMPRIL|LAN
                              ...
                          RIL|LANRNEVFHREIIS
                           IL|LANRNEVFHREIISE
                            L|LANRNEVFHREIISEM
```

The subpeptides are constructed as follows [*C++*]:

```
string line = "";

unordered_map<string, int> kmer_dict;

while(getline(in_stream, line)){

      istringstream iss(line);
```

```
string aa_full;
string line_label;
string nt_L;
int nt_L_size;
string nt_R;
int nt_R_size;
string aa_L;
int aa_L_size;
string aa_R;
int aa_R_size;

string aa_left = aa_L;
string aa_right = aa_R;

if(aa_L_size > (kmer_length-1)){
      int overage = aa_L_size - kmer_length + 1;
      aa_left = aa_L.substr(overage);
}
if(aa_R_size > (kmer_length-1)){
      aa_right = aa_R.substr(0,kmer_length-1);
}

string full_aa = aa_left + aa_right;
auto line_size = full_aa.size();

int start = 0;
int stop = line_size - kmer_length + 1;

while(start < stop){
      string current_kmer = full_aa.substr(start, kmer_length);
      ++kmer_dict[current_kmer];
      start++;
      }
}
```

Construct protein databases from proteome

To include as many reference proteins as possible in our reference proteome database [99] we start from the intersection of three publicly available human proteome references: RefSeq, UnitProt, and Gencode. We also use a common protein contaminants file available from the MaxQuant website. We download the reference proteomes in FASTA format and convert them to tbl files using AWK script [*Bash*]:

```
>> awk -f FastaToTbl.awk [fasta_file] > [tbl_file]
```

We then use the *C++* utility "build_protein_partitions" to split each proteome into four subsets (We compute the size for each of the four partitions for a proteome with 1003 proteins by rounding up 1003/4 to 251, so that we get partitions with sizes 251,251,251, and 250) [*Bash*]:

```
>> build_prot_parts [input_file] [output_file] [partition_size]
```

We also produce fixed-length peptide files for these partitions, by running the *C++* utility "build_kmers_simple" for each length *L* [*Bash*]:

```
>> build_kmers_simple [input.tbl] [output.tbl] [L]
```

To build overall reference proteome peptide files for each length *L* we use *Bash* functions cat, sort, and unique, then remove any peptides that include stop codon letters 'J', 'O', or "U' [*Bash*]:

```
>> cat [sub]/partition_*/kmers_[k].tbl > [sub]_all_kmers_[k].tbl
>> sort [sub]_all_kmers_[k].tbl > [sub]_all_kmers_[k]_SORTED.tbl
>> uniq kmers_[k]_SORTED.tbl > kmers_[k]_uniq.tbl
```

To remove stop codon letters we use the *Awk* expression '!/J|O|U/', which we can read as "does not contain any combination of the strings 'J' or 'O' or 'U' " so that lines from our input file are written to our output file only if they do not contain those letters [*Bash*]:

```
>> awk '!/J|O|U/' [sub]_kmers_[k].tbl > [sub]_kmers_[k]_clean.tbl
```

4. Search for Queries in Databases

The goal of this step is to determine which of the *de novo* predicted peptides are found in our protein search spaces and if so where they are found.

In the membership search we determine the subset of query kmers that are found in the protein partition's kmer list. The two input files are single-column files with no headers (.tbl files) containing for some peptide length the query peptides and the database peptides of that length. We use the Bash join command to compare the first column of the two input files ("-j 1") and output a file containing every entry in the first column of the query file ("-o 1.1") that is found in the first column of the db file (it's the intersection of the two peptide sets). The input

files must both be sorted (this was done in previous steps) and the output file retains the original ordering from the input files and so is sorted. [*Bash*]:

```
>> join -j 1 -o 1.1 query_kmers.tbl db_kmers.tbl > matches.tbl
```

In practice, the vast majority of query peptides will be absent from any given database partition and so the output file for this step is orders of magnitude smaller than the input query file.

In the location search we use the *C++* utility "search_query_in_ref" to find the precise locations of the peptide queries (matches.tbl) in the full database partition file (proteins.tbl). The query input file consists of a single column of peptide sequences (matches.tbl). The database input file consists of a tab-separated two-column file where each row provides a protein label followed by a protein sequence. The output is a file with two columns: the query peptide and the protein label for the protein sequence in which the query was found [*Bash*]:

```
>> search_query_in_ref matches.tbl proteins.tbl output.tbl
```

In the search function we load both input files into memory as string vectors, and then iterate through them, adding any matches to the results file, where match entries include the query sequence and the reference label (not the reference protein sequence). We build three vectors for the import [*C++*]:

```
vector<string> q_vector_original;
vector<string> ref_vector_label;
vector<string> ref_vector_seq;
```

We then iterate over these, so that for each query at position *i* we test whether it is found anywhere in the protein sequence at position *j* and if so, we print to our results file the query peptide at *i* and the protein label at position *j*. First, we search for query in protein sequence and assign the output to "found"; then we execute the contents of the if-statement if "found" is not equal to npos (meaning not found at any position, no position, in the string) [*C++*]:

```
for(int i = 0; i < query_size; i++){
```

```
        for(int j = 0; j < ref_size; j++){

              string q_find = q_vector_original[i];
              string ref_search = ref_vector_seq[j];
              std::size_t found = ref_search.find(q_find);

              if ( found != std::string::npos ){

                    results_stream << q_vector_original[i] << "\t"  \\
                                << ref_vector_label[j] << "\n";

              }
        }
}
```

Next, we aggregate the search results for each query set and database across partitions (one file for the membership searches and one file for the location searches). For the membership search matches we produce a single-column file with the list of all query peptides found in the database proteins for both the target and the decoy ('t/d') query sets [*Bash*]:

```
>> cat [t/d]/q_*_ref_*/q_hits.tbl > [out]/hits_[db]_[event]_[t/d].tbl
```

Since it is possible to have the same query peptide match in multiple protein partitions we sort and then find the unique set of each aggregate file [*Bash*]:

```
>> sort hits_[db]_[event]_target.tbl > sorted.tbl
>> uniq sorted.tbl > uniq.tbl
```

For the location search matches, we produce a two-column file with the list of all query peptides (first column) and the proteins in which they are found (second column) for both target and decoy query sets [*Bash*]:

```
>> cat [t/d]/ref_*/q_db_seqs.tbl > [out]/locs_[db]_[event]_[t/d].tbl
```

Finally, we combine the search results with the *de novo* PSP files by matching up the Peptide_TD entries in the former with the Peptide_simple entries in the latter.

5. Predict MHC Binding

We predict MHC binding values for our Peptide_IL set. First, we take the query set and we run it through the *C*++ "build_fasta_parts" utility to produce small FASTA files suitable for running small MHC binding sets on the quick partition. This utility takes as input the *n*-line single-column peptide file and produces fasta files with a peptide label line and a peptide sequence line for each set of 200 of the input peptides [*Bash*]:

```
>> build_fasta_parts [input_tbl] [output_dir] [part_size]
```

Where the FASTA [109] file looks, with the label line starting with '>', like:

```
>peptide_1
YTDATSKY
>peptide_2
SEFEKNKKL
```

Next we run NetMHC on each fasta partition (using the pvactools suite [110] on Biowulf) using the sample-specific HLA-1 alleles (2 each for HLA-A, HLA-B, HLA-C) [*Bash*]:

```
>> module load pvactools
>> pvacbind run input.fasta MD55A3 "HLA-A*02:05,HLA-A*03:01, \\
      HLA-B*07:02,HLA-B*40:32,HLA-C*06:02,HLA-C*07:02" \\
      {NetMHC} output_dir -e 8,9,10,11 --n-threads 32  \\
      --net-chop-method "cterm" --netmhc-stab
```

The output file contains the predicted binding strength for each peptide for each HLA allele (in addition to other data we do not use here).

6. Filter Peptide-Spectrum Matches

To construct our retention time models we match the retention times from the raw MS files to the PSPs in the *de novo* sequencing output. Since the LC-MS/MS experiments used a linear gradient we expect a linear relationship between this RT prediction and the actual RT. We are not interested in predicting the actual RT, but only in modeling the relationship between Pred_RT and RT so that we can exclude candidate peptides whose Pred_RT values are outliers. We use the index-based SSRC model to predict retention times from amino acid sequences based on

individual amino acid masses and charges, and a small set of sequence-dependent features [95].

We use the SSRC function from the protViz library [95] [*R*]:

```
unique_IL[,Pred_RT:=unlist(lapply(Peptide_IL_TD, ssrc))]
```

Then, for each experiment and each peptide length we build a linear model [*R*]:

```
retention_model = lm(Pred_RT ~ RT, data=temp_dt)
```

We extract three parameters from the model for later use [*R*]:

```
model_coeff=retention_model$coefficients
model_slope=model_coeff[2]
model_intercept=model_coeff[1]
model_summary = summary(retention_model)
model_adjrsq = model_summary$adj.r.squared
```

and compute the Pearson correlation between the RT and Pred_RT (with one-sided t-test) [*R*]:

```
pearson = cor.test(x=temp_dt$RT, y=temp_dt$Pred_RT, method="pearson",
alternative="greater")
pearson_R = pearson$estimate
pearson_p = pearson$p.value
```

The residual for any given Pred_RT is the difference between it and the model prediction for its Acual_RT,

$$residual = Pred\_RT - (slope \times Actual\_RT + intercept)$$

We remove outliers, defined as having model residuals that are either less than Q1-IRQ*1.5 or greater than Q3+IQR*1.5 where IQR is the difference between the first and third quartiles, Q3-Q1. (We pick these cutoffs because the residuals should be normally distributed.)

To compute our FDR Q scores for each sample-database search we first sort the PSMs by their PSM score ("AA_val_mean" in the *R* code) and compute the cumulative ratio of decoy to target identifications (FDR) starting from the highest PSM score. Next, we run a loop through the FDR scores to compute the FDR Q scores [*R*]:

```
setorderv(process, "AA_val_mean", order=-1)
process[,T_1:=ifelse(TD=="target", 1, 0)]
process[,D_1:=ifelse(TD=="decoy", 1, 0)]
process[,T_unit_score:=sum(T_1), by="AA_val_mean"]
process[,D_unit_score:=sum(D_1), by="AA_val_mean"]
```

```
to_return = unique(process[,.SD,.SDcols=c("AA_val_mean",
"T_unit_score", "D_unit_score")])
to_return[,T_score_cum:=cumsum(T_unit_score)]
to_return[,D_score_cum:=cumsum(D_unit_score)]
to_return[,FDR_score:=ifelse(T_score_cum==0, 1,
D_score_cum/T_score_cum)]
to_return[,FDR_score:=ifelse(FDR_score>1, 1, FDR_score)]

q_pile = data.table(AA_val_mean=integer(), Q_score=numeric())

for(score in unique(to_return$AA_val_mean)){
     temp_results = to_return[AA_val_mean>=score,]
     Q_score = max(temp_results$FDR_score)
     the_list = list(score, Q_score)
     q_pile = rbind(q_pile, the_list)
 }

 to_return = merge(to_return, q_pile, by="AA_val_mean")
 to_return= unique(to_return[,.SD,.SDcols=c("AA_val_mean",
      "Q_score", "FDR_score")])
 process = merge(process, to_return, by="AA_val_mean")
 process[,c("T_unit_score", "D_unit_score", "T_1", "D_1"):=NULL]
```

7. Quantify RNA Expression

To quantify RNA expression we retrieve the raw RNA-seq data, perform quality control, align the RNA-seq reads to our transcriptome reference and quantify gene expression. Whereas we run most steps on Biowulf, we are advised to run the raw RNA-seq file retrieval on Helix [*Bash*]:

```
>> module load sratoolkit/3.0.2
>> fasterq-dump -p -t /scratch/$USER SRR5088813
```

We then use Trimmomatic to remove low-quality read regions and any remaining PCR adapter sequences [*Bash*]:

```
>> module load trimmomatic/0.39
>> java -jar $TRIMMOJAR SE -phred33 -threads 4 {SAMPLE_FASTQ_FILE}
{SAMPLE_FASTQ_QC} ILLUMINACLIP:/usr/local/apps/trimmomatic/Trimmomatic-
0.39/adapters/TruSeq3-SE.fa:2:30:10 LEADING:3 TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:33
```

We found we lost ~10% of reads using a sliding window of 4:20, 2.5% with 4:15, and 0.01% with 4:10, so we decided to use 4:15.

Next we use RSEM [111] and STAR [112] to build a genome reference for the Gencode v43 transcriptome [*Bash*]:

```
>> module load STAR/2.7.9a
```

```
>> module load rsem/1.3.3
>> rsem-prepare-reference --gtf gencode.v43.annotation.gtf \\
      --star --star-sjdboverhang 85 --num-threads 64 \\
      GRCh38.p13.genome.fa /gencode_v43/gencode_v43_overhang85/v43
```

Then we align reads using STAR and quantify gene expression using RSEM [*Bash*]:

```
>> module load STAR/2.7.9a
>> module load rsem/1.3.3
>> rsem-calculate-expression --strandedness reverse \\
      --num-threads 24 --star --time {SAMPLE_qc.fastq} \\
      {gencode_v43/overhang85/v43}
```

Results

Although the pipeline is designed to analyze peptides of different lengths (i.e. those between 8 and 14 amino acids found in HLA peptidomics data), because the vast majority of predicted and identified peptides were nine amino acids in length we further analyze only peptides of this length to maximize our chances of filtering out noise and identifying rare events (this is commonly found to be the most abundant peptide length recovered in HLA-peptidomics studies).

Peptide-spectrum match filters

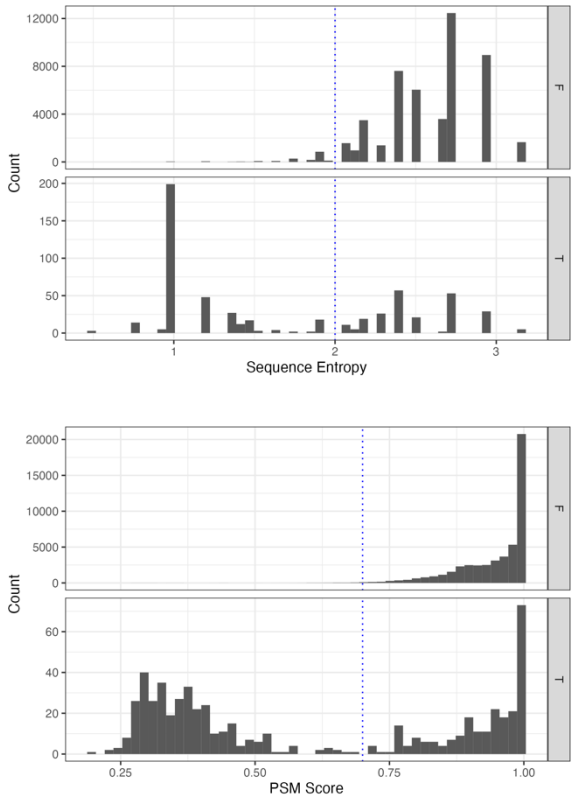
Retention time models

Most canonical peptide-spectrum matches were modeled well by the retention time (RT) model, and were unimodally distributed with respect to sequence entropy (**Figure 9**, top) and PSM score (**Figure 9**, middle). In contrast, the RT model outliers were bimodally distributed with respect to sequence entropy and PSM score. Based on these distributions we selected the entropy value of 2, which approximates the boundary between the two outlier distributions, as a minimum entropy filter. Likewise, we chose the PSM score value of 0.70, which approximates the boundary between the two outlier distributions, as a minimum PSM score filter. Across all canonical

peptide-spectrum matches sequence entropy and PSM score roughly cluster (**Figure 9**, bottom) into two groups.

Peptide singleton filter

We found that filtering out predicted peptide sequences that were predicted for only one sample greatly improved FDR control without greatly reducing the total numbers of confident peptides identified (see below). Accordingly for each database search we show results with and without this peptide singleton filter.



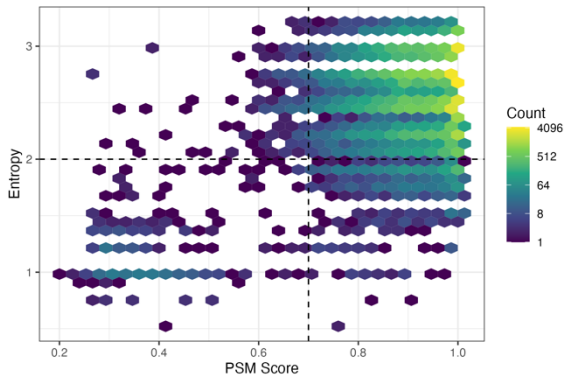


Figure 9. Canonical database retention time models.
Model outlier peptides are bimodally distributed with respect to sequence entropy (top) and peptide-spectrum match scores (middle). The heatmap (bottom) shows how entropy as a function of PSM score is also bimodally distributed, with most PSMs having both a high PSM score and a high sequence entropy. We chose a log2 scale for the heatmap color gradient to better visualize smaller values. Dotted lines indicate entropy cutoff of 2 (top plot, vertical line; bottom plot, horizontal line) and PSM score cutoff of 0.7 (middle, vertical line; bottom plot, vertical line).

RNAseq filter

From the four RNA-seq samples (SRR10814112-5) (NT,NT,IFNg,IFNg) for expression above 0 TPM, the union of all four gave 80683 of 252913 transcripts (31.9%) and 21970 of 62703 genes (35.0%). For expression above 1 TPM, 40715 transcripts (16.1%) and 14123 genes (22.5%). Filtering peptides by source gene expression (either nonzero or above 1 TPM) or source transcript expression (either nonzero or above 1 TPM) reduced the total numbers of peptides examined and the total numbers of confident peptides identified, but had little impact on FDR control (not shown). Accordingly, we did not use RNA-seq expression as a filter.

Target vs. Decoy Comparison Metrics

PSM score vs. FDR Q Score

We plot for each analysis the FDR Q score as a function of PSM score for each of the twelve samples to visualize the impacts of sample condition, PSM filters, and database on FDR control. We also provide for each analysis a table with counts of target peptides passing FDR Q<=5%.

NetMHC Predicted Binding Strength for Targets vs. Decoys

We plot for each analysis the across-sample distribution of peptide minimum NetMHC binding score percentiles for target and for decoy PSMs. Since we expect NetMHC binding scores percentiles for true target peptides to be strong (recall that a lower predicted IC50 and thus a lower binding score percentile indicates stronger binding) compared to those for decoy peptides we visualize the impacts of our PSM filters on these distributions using the distribution of scores for 10E4 randomly selected canonical database peptides (**Figure 10**) as a basis for comparison. If our filters are effective we expect filtering to make our peptide score distributions more peaked and more right-skewed.

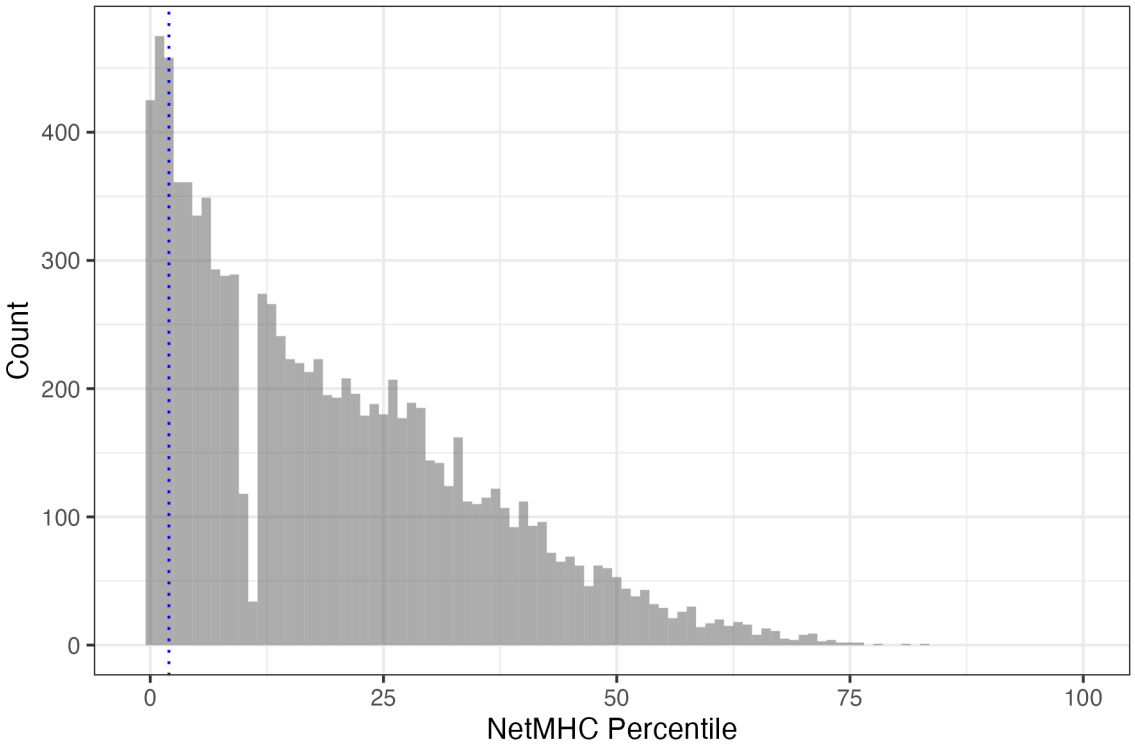


Figure 10. Distribution of top NetMHC percentile score for each of 10E4 randomly selected peptides of length 9 from the canonical database.
The NetMHC algorithm predicts binding strength between a peptide and a specific HLA-1 receptor (specific to the HLA-A, B, or C allele coding the receptor). The binding strength is reported in terms of nanomolar IC50, the concentration of peptide required for the copies of the peptide to be bound to 50% of available HLA receptors. A lower IC50 indicates a stronger binding strength. The NetMHC percentile score gives the percentile for a given HLA binding strength when compared to a background distribution of binding strengths for peptides of the same length binding to HLA receptors of the same allotype. As a matter of convention, a pair with a predicted binding strength percentile at or below 0.5% is predicted to have "strong binding", whereas at or below 2% it is predicted to have (at least) "weak binding". The blue dotted vertical line in the plot is positioned at 2%.

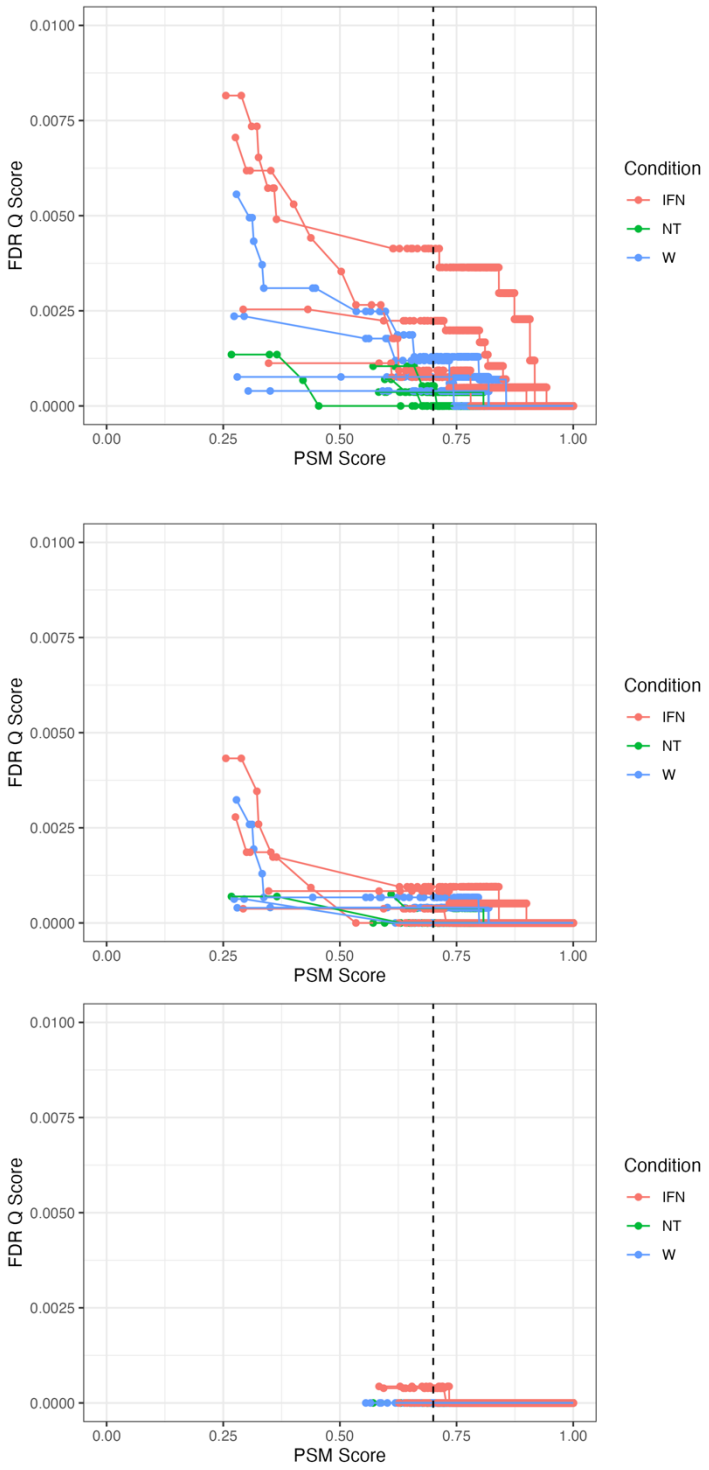


Figure 11. FDR Q Score versus PSM score for canonical database searches.
The top plot shows results without the singleton or entropy filters applied. The middle plot shows results with the singleton filter applied. The bottom plot shows results with both the singleton and entropy filters applied.

Canonical peptide database searches

Canonical database searches identified between 1034 and 2853 target peptide spectrum matches with PSM ≥ 0.7 and passing the FDR Q score cutoff of 5% (**Table 6**). In fact, the maximum FDR Q score for these matches across conditions and filters was less than 0.5% (**Figure 11**). Samples of the IFNg treatment condition tended to have higher FDR Q scores than samples of the W-depletion or no treatment conditions. Across sample conditions, FDR Q scores were lowered by addition of the singleton filter and again lowered by addition of the entropy filter.

In the no-filter condition, the distribution of MHC binding predictions for target PSMs was more peaked and more positively skewed than the distribution of MHC binding predictions for decoy PSMs (**Figure 12**). The decoy distribution is much closer than the target distribution to the MHC binding score distribution for randomly selected canonical database peptides (**Figure 10**). Distributions for the singleton filter and singleton plus entropy filter condition are similar for the target PSMs. However, very few decoy PSMs passed these filters, making their distributions less informative.

Table 6. Target PSMs passing FDR Q score ≤ 5% for canonical searches by sample and filters.

Sample	Filter(s)		
	None	Singleton	Singleton + Entropy
NT1	2853	2650	2569
NT2	2732	2601	2531
NT3	1908	1803	1749
NT4	1473	1433	1397
W1	2635	2543	2463
W2	2743	2608	2526
W3	1601	1533	1491
W4	1685	1585	1544
IFN1	2659	2398	2311
IFN2	3140	2690	2593
IFN3	1210	1146	1117
IFN4	1115	1065	1034

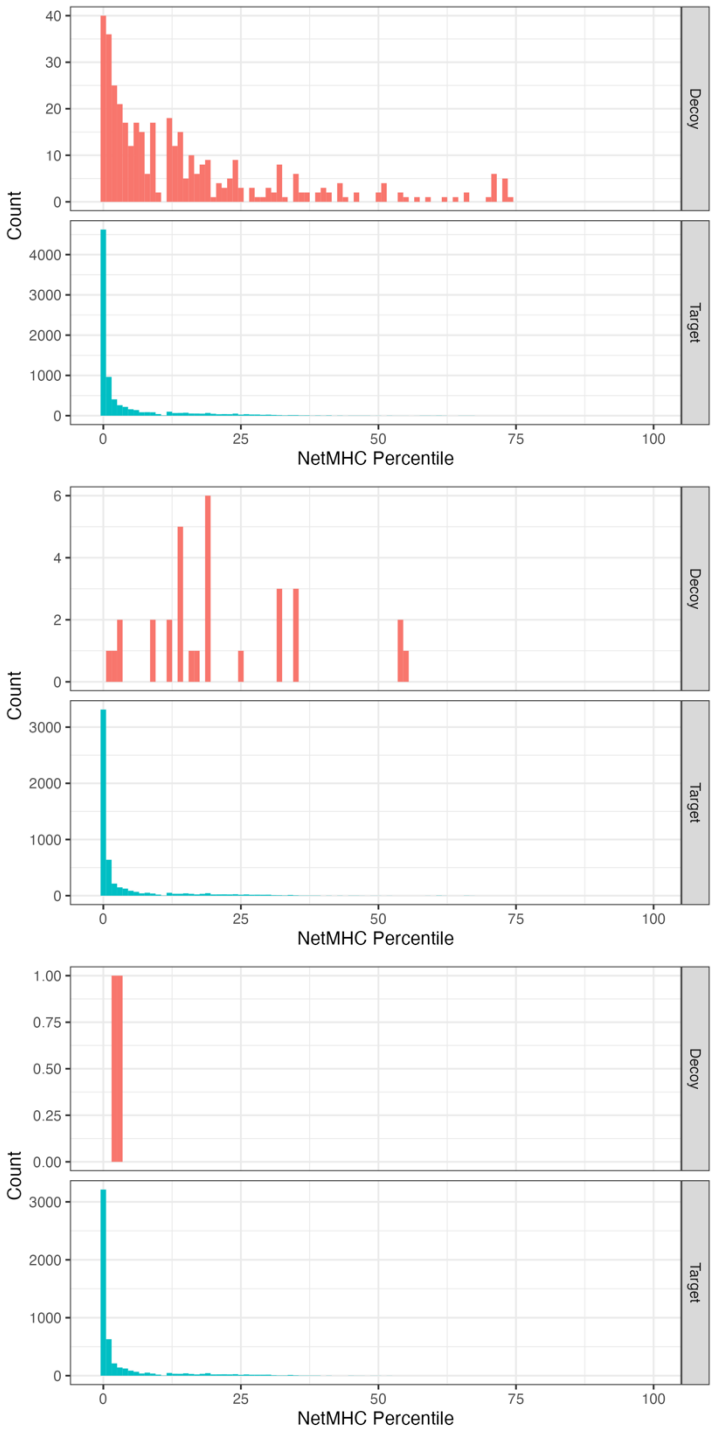


Figure 12. Distributions of predicted MHC binding scores for canonical database PSMs. The top plot shows results without the singleton or entropy filters applied. The middle plot shows results with the singleton filter applied. The bottom plot shows results with both the singleton and entropy filters applied.

Gencode noFS peptide database searches

Gencode noFS database searches identified between 51 and 261 target peptide spectrum matches with PSM ≥ 0.7 and passing the FDR Q score cutoff of 5% (**Error! Reference source not found.**). Samples of the IFNg treatment condition tended to have higher FDR Q scores than samples of the W-depletion or no treatment conditions. Across sample conditions, FDR Q scores were lowered by addition of the singleton filter and again lowered by addition of the entropy filter (**Figure 11**).

In the no-filter condition the distribution of MHC binding predictions for target PSMs was more peaked and more positively skewed than the distribution of MHC binding predictions for decoy PSMs (**Figure 14**). The decoy distribution is much closer than the target distribution to the MHC binding score distribution for randomly selected canonical database peptides (**Figure 10**). Distributions for both target and decoy PSMs became more peaked and positively skewed with the addition of the singleton filter and to a smaller extent with the addition of the singleton plus entropy filter.

Table 7. Target PSMs passing FDR Q score ≤ 5% for noFS searches by sample and filters.

Sample	Filter(s)		
	None	Singleton	Singleton + Entropy
NT1	233	227	210
NT2	194	189	202
NT3	172	169	163
NT4	119	127	133
W1	286	261	239
W2	259	237	242
W3	172	184	173
W4	192	184	181
IFN1	128	114	137
IFN2	114	96	154
IFN3	55	53	91
IFN4	31	51	66

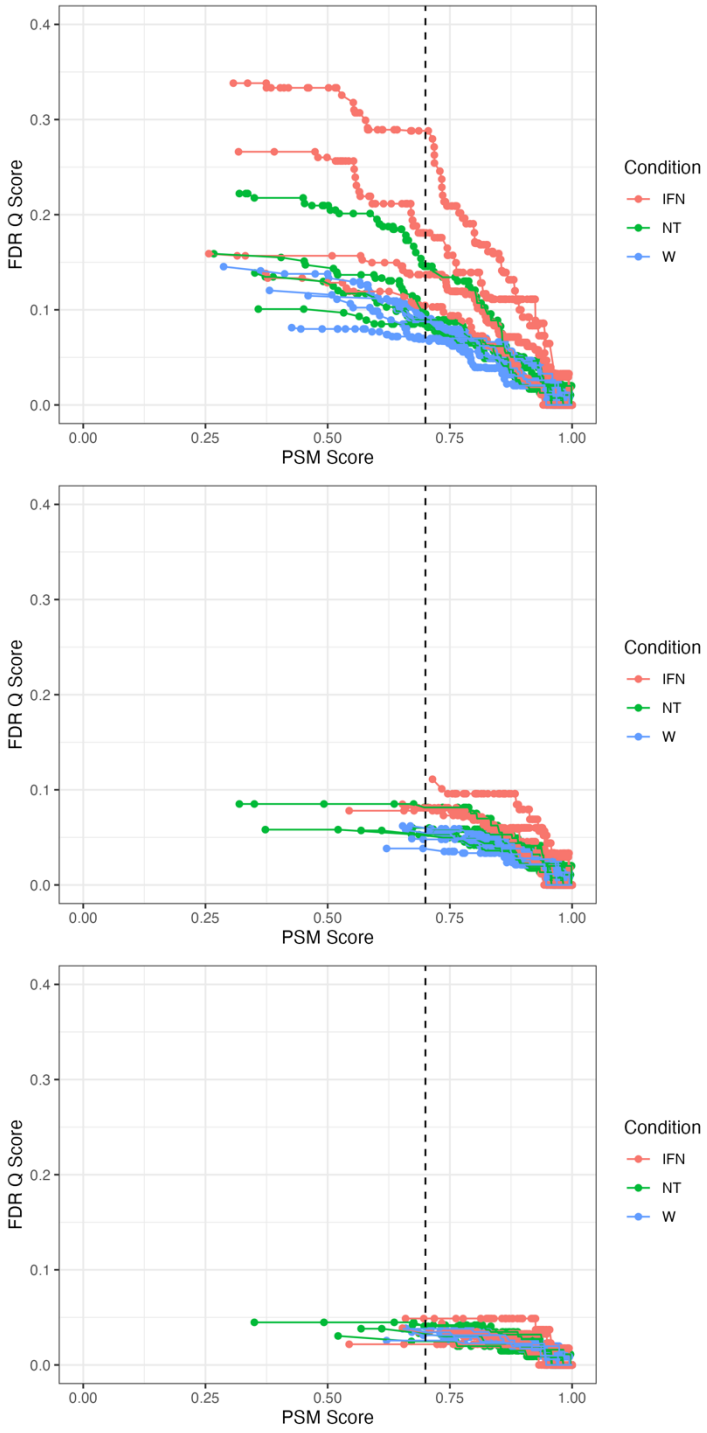


Figure 13. FDR Q Score versus PSM score for noFS database searches.
The top plot shows results without the singleton or entropy filters applied. The middle plot shows results with the singleton filter applied. The bottom plot shows results with both the singleton and entropy filters applied.

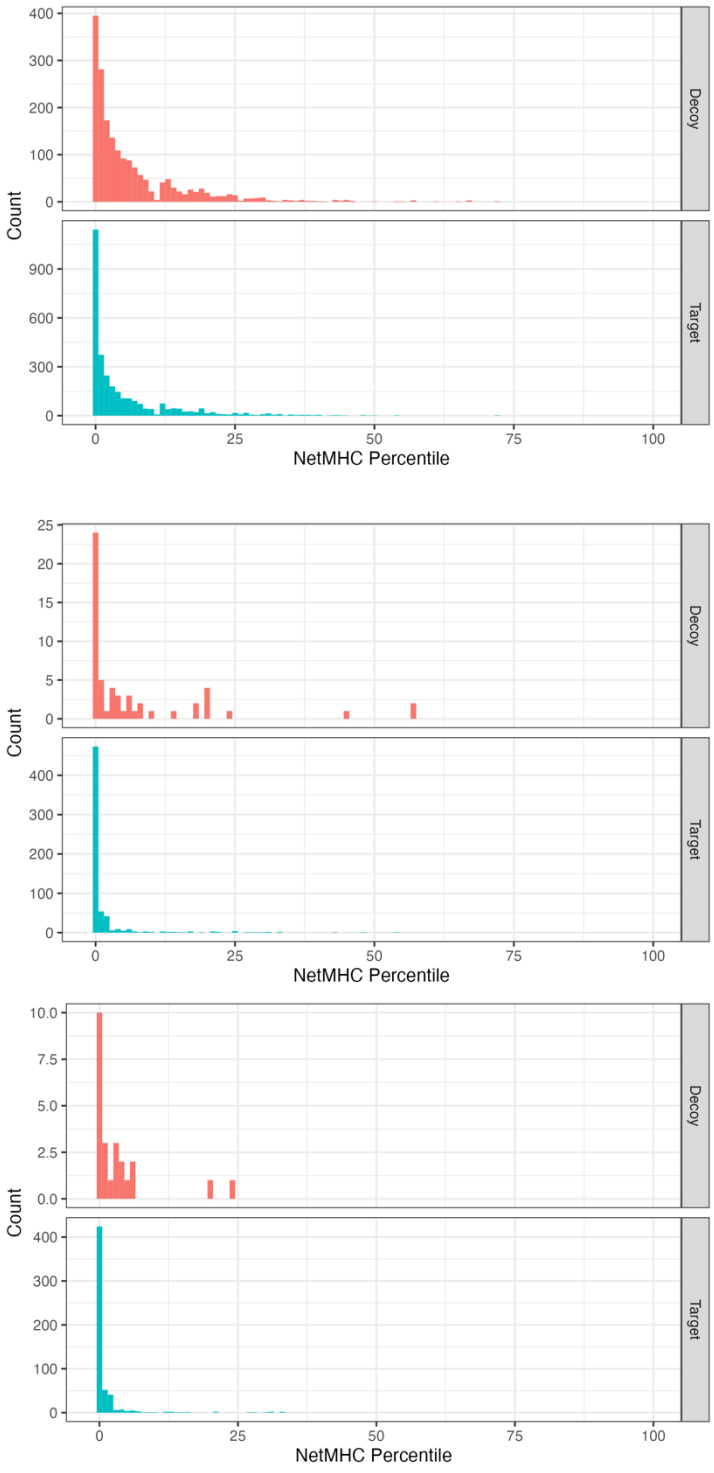


Figure 14. Distributions of predicted MHC binding scores for Gencode noFS database PSMs. The top plot shows results without the singleton or entropy filters applied. The middle plot shows results with the singleton filter applied. The bottom plot shows results with both the singleton and entropy filters applied.

Gencode FSud peptide database searches

Gencode FS database searches identified between 0 and 8 FSu and either 0 or 1 FSd target peptide spectrum matches with PSM ≥ 0.7 and passing the FDR Q score cutoff of 5% (**Table 8**, **Table 9**). Across sample conditions for the FSu events, FDR Q scores were lowered by addition of the singleton filter and again lowered by addition of the entropy filter (**Figure 15**). For FSd events FDR Q scores remained high regardless of filter conditions, with a total of four events passing the Q score cutoff across the twelve samples.

In the no-filter condition the distribution of MHC binding predictions for target PSMs was similarly peaked and skewed with respect to the distribution of MHC binding predictions for decoy PSMs (**Figure 14**). The no-filter target and decoy distributions are much closer to the MHC binding score distribution for randomly selected canonical database peptides (**Figure 10**) than the filter condition distributions. Distributions for both target and decoy PSMs became more peaked and positively skewed with the addition of the singleton filter and to a smaller extent with the addition of the singleton plus entropy filter.

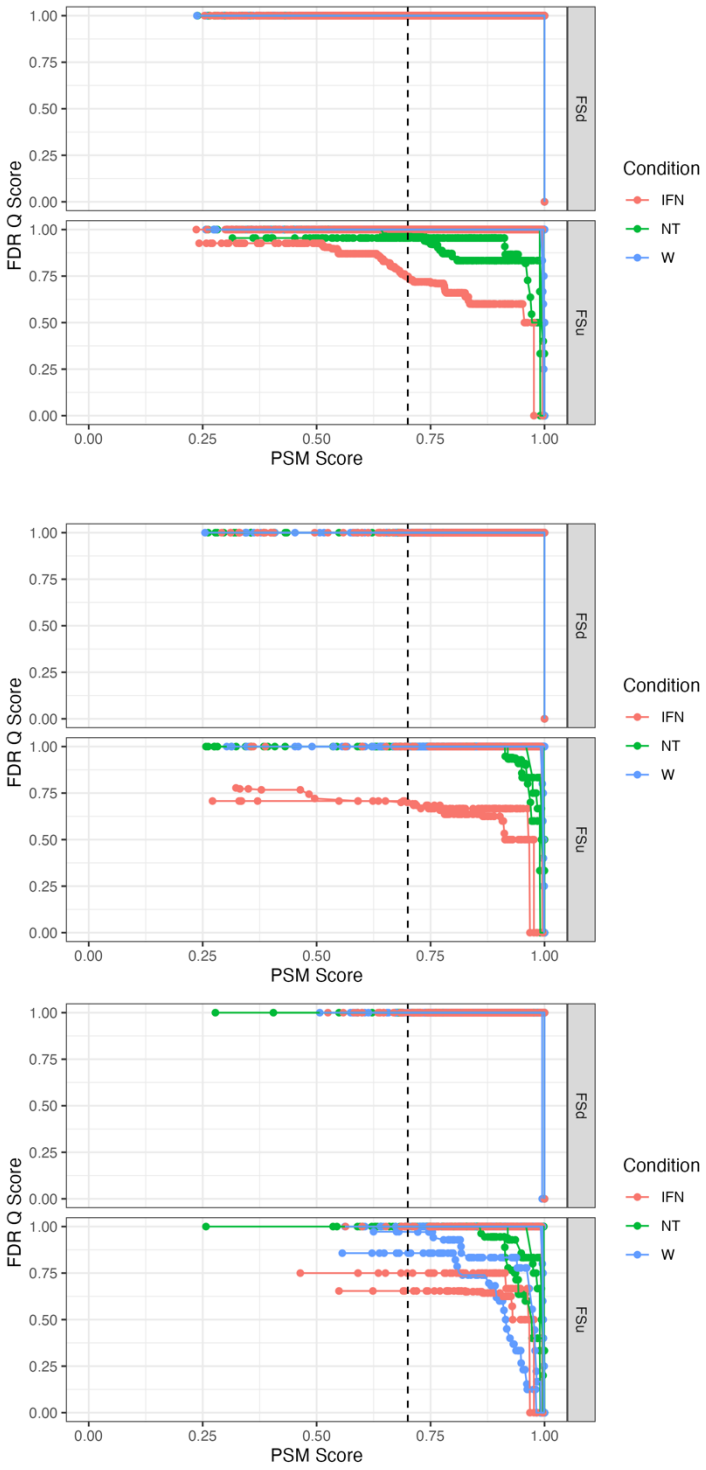


Figure 15. FDR Q Score versus PSM score for FSud database searches.
The top plot shows results without the singleton or entropy filters applied. The middle plot shows results with the singleton filter applied. The bottom plot shows results with both the singleton and entropy filters applied.

Table 8. Target PSMs passing FDR Q score ≤ 5% for FSu searches by sample and filters.

Sample	Filter(s)		
	None	Singleton	Singleton + Entropy
NT1	2	2	5
NT2	3	3	3
NT3	1	1	1
NT4	3	3	3
W1	2	4	4
W2	4	4	4
W3	0	0	6
W4	0	0	8
IFN1	0	0	0
IFN2	1	1	1
IFN3	2	2	2
IFN4	0	2	2

Table 9. Target PSMs passing FDR Q score ≤ 5% for FSd searches by sample and filters.

Sample	Filter(s)		
	None	Singleton	Singleton + Entropy
NT1	0	0	0
NT2	1	1	1
NT3	0	0	0
NT4	0	0	0
W1	0	0	0
W2	1	1	1
W3	0	0	1
W4	0	0	0
IFN1	0	0	0
IFN2	1	1	1
IFN3	0	0	0
IFN4	0	0	0

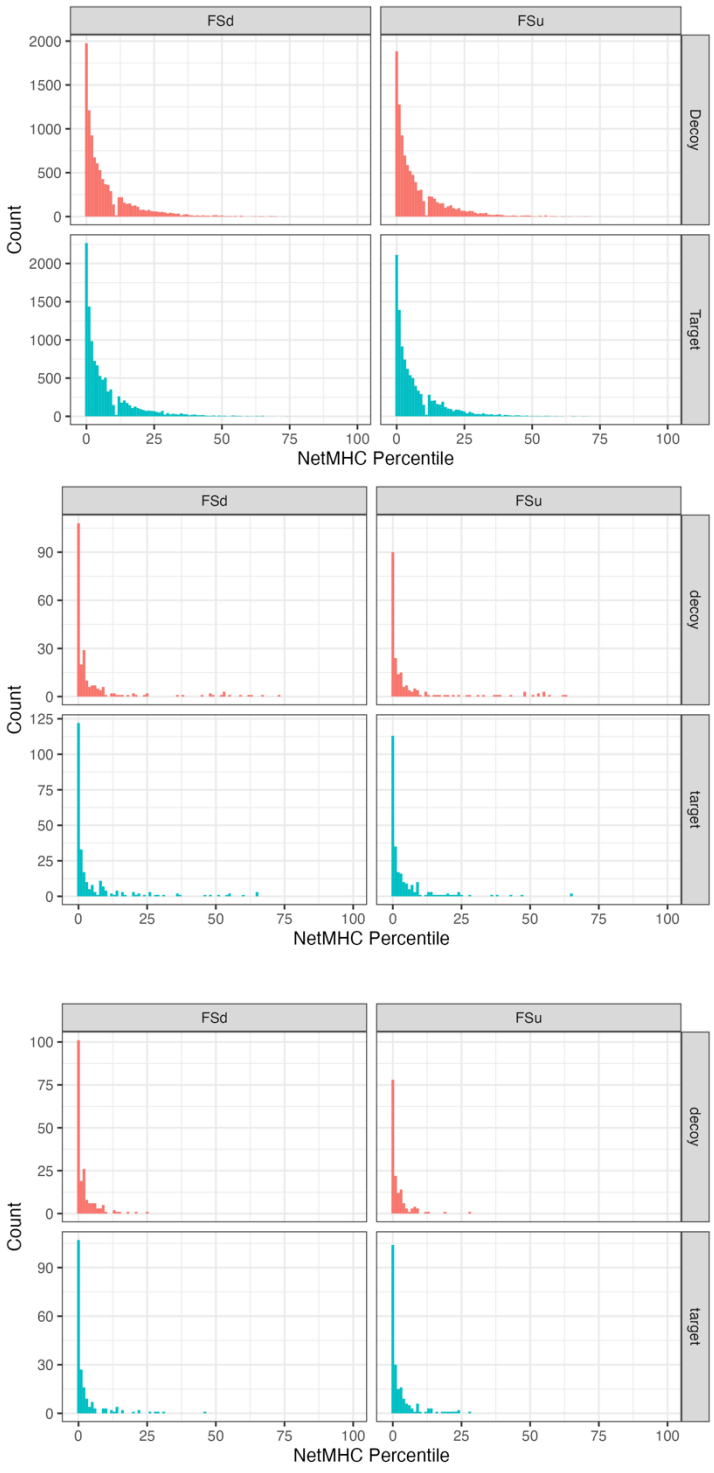


Figure 16. Distributions of predicted MHC binding scores for frameshift (FSu, upstream; FSd, downstream) database PSMs.
The top plot shows results without the singleton or entropy filters applied. The middle plot shows results with the singleton filter applied. The bottom plot shows results with both the singleton and entropy filters applied.

Discussion

FDR control and MHC binding enrichment

Our results demonstrate that we can control FDR to some extent for searches of our constructed databases. As expected, FDR control was best for the canonical database, less effective for the larger noFS database, and even worse for the even larger FSu and FSd databases. MHC binding enrichment differences between our target and decoy sets and changes in these enrichments resulting from additional singleton and entropy filters suggests that with more filters or more carefully constructed and consequently smaller databases or both we could achieve much better FDR control.

Frameshifts and motifs

Our analysis identified at total of 39 FSu peptides and 4 FSd peptides across our twelve samples. There was no clear qualitative pattern with respect to sample conditions, and the confounding effects of tryptophan depletion or IFNg treatment on levels of overall protein translation in the cell [87] make it difficult to quantitatively compare the detected events across conditions. None of the identified frameshift events occurred at a canonical slippery site [90, 89] and only one event occurred at a tryptophan codon.

Setting aside the different treatment conditions, the greater within-sample abundance of FSu events compared to FSd events we observe across samples may reflect a biological difference between the frequencies of FSu and FSd events. If we do not expect many FSd events to occur at all then we might consider the search for FSd events as a negative within-sample control for the search for the more common FSu events. In this case the rarity of identified FSd events compared to identified FSu events lends some credence to our identified FSu events.

Future directions.

Future iterations could improve the current approach in several ways. First, PSMs could be assessed not only in terms of sample-level retention time models but also in terms of the fragmentation coverage for each individual PSM as others have done [86]. Analyzing the number of predicted cleavage fragments for a peptide that are found in an MS2 scan (**Figure 5**, bottom) could help us filter out more low-quality PSMs. (The de novo sequencing tools explored for the current project do not provide fragmentation coverage analytics in their outputs, but it is possible that either their deep learning models or other analyzed metrics capture some of the information represented by a fragmentation coverage metric). Second, the simplistic *in silico* translation strategy employed here could be modified to take into account translation initiation sites, whether the canonical "AUG" or other less common motifs [113]. Likewise, we could allow for stop-codon readthrough [107]. Peptides resulting from alternative initiation site usage or stop-codon readthrough or both might be flagged so that the user could analyze this database separately from other databases. These steps could help reduce the sizes of our databases. Third, we could compare outputs resulting from multiple *de novo* MS sequencing programs or multiple MHC binding prediction programs. Such an approach could allow different tools to complement each other or at least highlight when one tool clearly outperforms another. Finally, we could apply the pipeline (not including MHC binding prediction) to whole proteomics data. This would be advantageous because whole proteomics outputs have many more peptide spectra than immunopeptidomics outputs and thus we might have more success in detecting rare events. The whole proteomics files for the Bartok et al. study from which we retrieved our immunopeptidomics files have about twenty times more spectra for each sample

than the immunopeptidomics files. We ran most of the current pipeline for these whole proteomics files but ran into scaling issues such that we will need to modify several of the modules to accommodate the increased data magnitude.

Conclusion

We have demonstrated that a large-scale search for aberrant translation event products in immunoproteomics data is feasible. This allows us to perform transcriptome-wide unbiased searches such that we can study the downstream effects of translational perturbations without pre-determining restricted subsets of events for our database. False discovery rate control is possible for larger and less target-rich databases, but could be greatly improved. We have identified several additional straightforward features we might add to improve our ability to confidently identify novel events in proteomics data.

Chapter 3: Multi-level Gene Expression Survival Analysis for Cancer Gene-of-Interest Discovery

Introduction

Survival analyses use patient data to identify potential associations between clinical characteristics and survival time. A popular approach in cancer research to identify genes of interest for further investigation is to model patient survival as a function of gene expression, age, and sex. The status quo gene expression metric for these survival analyses is gene-level RNA-seq expression level. However, the roles of functional changes in splicing isoform expression can play in cancer biology are increasingly being recognized [13] and some researchers have published survival analyses using the metric of isoform abundance with the aim of identifying genes of interest not found when using the metric of gene abundance [19, 20, 21, 22]. In this chapter we analyze bulk RNA-seq data from TCGA melanoma samples using hierarchical multivariate Cox proportional hazards survival models to examine the predictive value of multivariate models based on different below-gene-level expression metrics in addition to subject age and sample purity. We hypothesize that some fitted models using below-gene-level metrics will identify survival associates not identified by fitted gene-level models, thus increasing the exploratory reach of gene expression-based cancer survival models.

Gene expression metrics

Our analyses use bulk RNA-seq data pre-processed and quantified at the gene and spliced isoform levels in transcripts-per-million (TPM) units [114, 115]. We further process this data to quantify expression in terms of: isoform-to-gene abundance ratio (isoR); alternative splicing

event percent-spliced-in (PSI); logit-transformed PSI (logitPSI); and splicing event TPM (eventT).

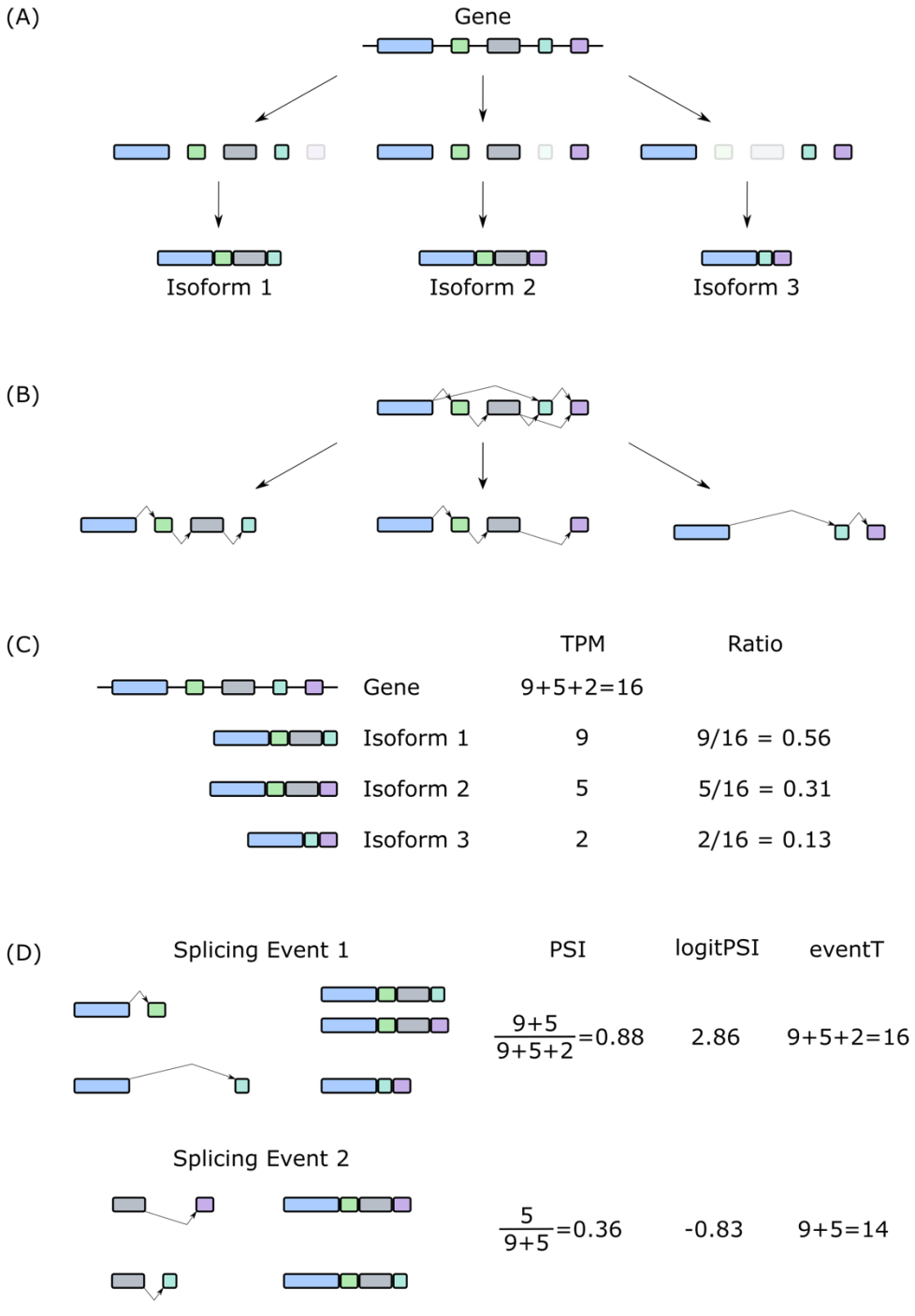


Figure 17. mRNA splicing and metrics. The simplified example gene has five exons (colored rectangles) and three possible isoforms.

After a protein-coding gene is transcribed to form a pre-mRNA transcript the spliceosome complex cuts and re-joins exons at splice sites to form an mRNA transcript. Through splicing a single gene can produce numerous different mRNA isoforms, each including a different subset of the full gene's sections (**Figure 17** shows a simplified example gene). We can represent the possible splicing events for a gene as a directed acyclic graph with exons as nodes and splicing junctions as edges (**Figure 17**B). On this graph forks represent alternative splicing events. The splicing events producing a given isoform together constitute a single path through the splicing graph, but some paths on the graph might not correspond to actually occurring spliced isoforms.

We can use the pre-quantified gene-level TPM (geneT) and isoform-level TPM (isoT) values to compute our remaining three metrics. These metrics differ from the TPM metrics in that they are expressed as fractions and so will be undefined if the abundance of the denominator term is zero. Accordingly, even though a sample will have isoT=0 and geneT=0 for a gene it does not express, the isoR, PSI, and logitPSI values for its isoforms and splicing events will be undefined. For a given isoform the isoform ratio (isoR) equals the isoform TPM divided by the total gene TPM:

$$isoR = isoT/geneT$$

where geneT>0 (**Figure 17**D). Splicing event percent-spliced-in (PSI) is equal to the abundance of transcripts that include the measured alternative event divided by the total abundance of transcripts that include either of the alternative events (**Figure 17**D):

$$PSI = \sum isoT_m / \sum isoT$$

where $isoT_m$ is the abundance of an isoform which includes the measured alternative, $isoT$ is the abundance of an isoform which includes either of the alternatives in the splicing event, and the

sample expresses at least one isoform for this event so that $\sum isoT$>0. Splicing annotation conventions determine which of two alternative splicing options is the measured alternative for a particular event [116]. For such an event the event TPM (eventT) is simply the denominator used to calculate PSI, $\sum isoT$. Researchers often use a logistic transformation of PSI to make the distribution more suitable for models requiring near normally-distributed variables:

$$logitPSI = \log_b((PSI + a)/(1 - PSI + a))$$

for some base *b* (we use b=2) and some small positive value a (we use a=0.001). This transforms the [0,1] bounded PSI values, which often have a J- or U-shaped distribution, so that the logitPSI are normally distributed [117].

Survival models

We use survival analyses to investigate relationships between data collected at cancer diagnosis and how long a patient survives. Survival analysis involves statistical methods that account for a special issue with survival data --- some patients die and for the rest we know the lower limit on their survival time since they either leave the study before it ends or they survive until at least the end of the study (this is "right-censored" data).

Patient data

For our survival analysis we use the following patient and sample data from The Cancer Genome Atlas (TCGA) skin cutaneous melanoma (SKCM) dataset [114]: tumor sample bulk RNA-seq data quantified in at the gene and splicing isoform levels; tumor type (primary or metastatic); and patient clinical data including patient sex, age at diagnosis, and time to survival event (where the

event is either death or leaving the study). We also use tumor sample purity estimates computed from the sample RNA-seq data [118].

Cox models

We employ standard multivariate Cox proportional hazards models [119, 120, 121]. When we fit such a model we aim to maximize its likelihood, the probability of observing the data based on the fitted model parameters. A Cox proportional hazards (Coxph) model is based on a linear regression of a hazard rate on one or more covariates like gene expression or age. The hazard rate is the risk of experiencing an event (here, patient death) given that a patient has survived up to some time *t.* For example, the model

$$h_{t,i} = h_0 \cdot \exp\left(\beta_{age} \cdot age_i\right)$$

estimates the hazard rate for an individual *i* at time *t* as the baseline hazard rate h_0 (estimated based on survival times but not on any other individual data) multiplied by the natural exponentiation function of the subject's age multiplied by the model's age regression coefficient β_{age}. The model is semi-parametric because although the covariates are assumed to have a multiplicative effect on the hazard rate, the baseline hazard rate is estimated nonparametrically (it makes no assumptions about the statistical distribution of survival times). The model assumes constant proportionality between the hazards of any two individuals over time (such that, for example, the survival curves for two groups do not cross). A major advantage of the Cox model is that in addition to regressing over multiple variables it provides effect sizes: the hazard rate associated with a variable i is e^{β_i}, so that if the hazard rate is greater than 1 then increases in the variable increase the probability of death and if the hazard rate is less than 1 then increases in the variable decrease the probability of death. In addition to the regressing on quantitative variables,

our models stratify subjects to take patient sex into account. For stratified models, the fitted variable coefficients are equivalent for the different classes (male and female) but each class is assigned its own baseline hazard rate (h_0).

Comparing models

We assess the predictive power of a Coxph model by comparing its loglikelihood (LL) value to that of a simpler, null model. When we test a model such as

$$h_{t,i} = h_0 \cdot \exp\left(\beta_{age} \cdot age_i\right)$$

we compare it to a simpler model, in this case a model where the age coefficient is set to zero (and thus the age value does not affect the model):

$$h_{t,i} = h_0 = h_0 \cdot \exp(0 \cdot age_i)$$

We compute the likelihood ratio test (LRT) score,

$$LRT = -2 \times (LL_{fixed} - LL_{free})$$

where the simpler model is 'fixed' because it has a coefficient set to zero that in the more complex model is 'free' to change according to the value of the variable in question.

When we compare these models using a likelihood ratio test we are asking whether the model with the fitted value (i.e. the value that was free to fit the data) for the coefficient explains the data better than the model with the fixed value to an extent not likely by chance. We employ a Chi-squared test, with one degree of freedom since we have one variable fixed in the simple model but free in the complex model, to determine this probability from our LRT score.

<u>Published below-gene-level studies</u>

Three recent studies applied survival analysis methods to explore associations between changes in splicing isoform expression and patient survival in cancer. Zhang et al. applied a univariate

Cox model (regressing survival on isoform expression level) to TCGA data for thirty-one cancers [21]. Shen et al. applied a modified univariate Cox model that incorporates information about expression level uncertainty to study associations between splicing isoform ratios (specifically, the percent-spliced-in for particular exons) and patient survival in TCGA breast cancer patients [22]. West et al. performed log-rank survival analyses using empirically estimated null probabilities on TCGA lung cancer samples [20]. Chang et al. performed univariate Cox modeling to identify splicing events of interest in TCGA lung adenocarcinoma samples [19]. Although multivariate regression models taking into account factors like age and estimated sample purity in addition to expression levels have been used in gene-level survival analyses [122] to our knowledge this has not yet been published for cancers at the splicing isoform level.

In summary, our approach is novel in three ways. First, it employs a multivariate Coxph model, allowing us to compare models with a single expression level to those with two expression levels (e.g. comparing a model with geneT to a model with geneT+isoT) and to take into account subject age and sample purity. Second, our analysis compares models with different below-gene-level metrics: isoR, isoT, and PSI. These features allow us to explore the extent to which models including below-gene-level metrics with or without gene-level metrics are more predictive than those including gene-level metrics, and also the how models including one below-gene-level metric compare with those including a different below-gene-level metric. Third, to our knowledge this is the first study using the isoR and eventT metrics to study alternative splicing.

Methods

Data

Cancer patient data

For our survival analysis inputs we use the following patient and sample data sources: (1) tumor sample bulk RNA-seq data; (2) patient clinical data; and (3) tumor sample purity estimates. We use the associated clinical data for this study provided by the UCSC Xena project as well as the Broad TCGA GDAC Firehose project [123]. We use tumor sample purity estimates from [118]. In addition to the tumor sample RNA expression data and purity estimates we use the following variables from the patient and sample metadata: patient age at time of biopsy (Age); patient sex (male or female, Sex); tumor sample type (primary or metastatic); and patient survival time in days (TIME_TO_EVENT), which is coupled with an EVENT variable with the value "0" if the patient left the study at the given time or the value "1" if the patient died at the given time.

Gene expression

We use RNA-seq expression data from The Cancer Genome Atlas (TCGA) cutaneous melanoma (SKCM) study [114] for which the UCSC Xena project provides gene- and isoform-level expression quantification files ("TcgaTargetGtex_rsem_gene_tpm.gz" and "TcgaTargetGtex_rsem_isoform_tpm.gz" downloaded from the Xena server) [115]. Genes and isoforms are labeled by their Ensembl unique identifiers, and samples by their unique TCGA identifiers. Both files give expression as log2(TPM+0.001), which we convert back to TPM. We study only genes with positive expression in at least 25% of the SKCM samples we use.

Gene annotation

The TCGA data was quantified using the Gencode version 23 comprehensive (hg38) gene annotation ("gencode.v23.annotation.gtf.gz" downloaded from the Gencode server) [105]. The full annotation includes 60,498 genes with 198,619 transcripts isoforms. We use the annotation "Transcript_type" field to select protein-coding genes ("protein_coding"). This leaves us with 19,645 genes (each with at least one protein-coding transcript) with 143,483 transcript isoforms.

Subject age

We use a custom script to extract age values for each subject from the Firehose metadata files because the data file does not have a consistent column format (i.e. subject rows with no value for a column simply skip that column instead of filling it with a blank or "NA" value, such that the age value is in a different position for different sample rows). We run the following loop [*R*]:

```
for(age_cancer in age_cancers){
  open_me = paste0("Data/broad/gdac.broadinstitute.org_",age_cancer,
      ".Clinical_Pick_Tier1.Level_4.2016012800.0.0/All_CDEs.txt")
  bob = fread(open_me, header=FALSE)
  bob2=transpose(bob)
  name = c()
  age = c()
  found = 0
  row_num = 1
  while(found<2){
    row = unlist(bob2[[row_num]])
    if(row[1] == "bcr_patient_barcode"){
      name = row[2:length(row)]
      found = found+1
    }else if(row[1] == "age_at_initial_pathologic_diagnosis"){
      age = row[2:length(row)]
      found = found+1
    }
    if(row_num>100) break
    row_num = row_num+1
  }
  temp_dt = data.table(TCGA_Id=name,Age=age)
  master_age = rbind(master_age, temp_dt)
}
```

Survival data

We extract the objective survival time (OS) from the TCGA metadata for each subject. This time is given as days from diagnosis to event (where event is either leaving the study or death) and is a recommended endpoint for survival analysis of TCGA SKCM subjects (for some cancers, some of the standard four survival endpoints like objective survival or progression-free survival may not be appropriate) [124]. In order to avoid tests that are likely to lack sufficient statistical power, for our survival models we exclude models with expression metrics (isoR, PSI, and logitPSI) that are not defined for at least ten subjects with death events (EVENT=1) for each variable in the model [120] (This explains the lower bounds of 30 or 40 events for some of the model types; see **Table 10**.)

Tumor sample purity

We use tumor purity estimates from a pan-cancer TCGA tumor purity study (Supplementary Data 1 "Tumor purity estimates for TCGA samples. Tumor purity estimates according to four methods and the consensus method for all TCGA samples with available data") [118]. Of the five purity estimates available (including ESTIMATE, ABSOLUTE, LUMP, IHC, and a consensus value based on the previous four) we use values from the ESTIMATE method, which uses expression of immune and stromal genes to estimate the fraction of the bulk sample that is tumor cells.

Splicing computation

We run SUPPA2 [125, 116] (using Python 3.9.6) to compute percent-spliced-in (PSI) values for gene expression. We start with a TSV whose columns are Transcript_Id followed by samples and whose rows are TPM values. SUPPA2 takes as input a file of this structure, but missing the first

column label. To generate event definitions from our genome annotation file we use the SUPPA2 'generateEvents' function (*Bash*):

```
>> python3 suppa.py generateEvents \
      --input-file gencode.v23.annotation.gtf \
      --output-file gencode_v23_events --format ioe \
      --event-type SE SS MX RI AF AL A5 A3
```

Here we specify the output file header, the format we want as output (we want "ioe" which gives us local events) and the event types to include (SE, Skipping exon; MX, Mutually exclusive exons; A5, Alternative 5' splice-site; A3 Alternative 3' splice-site; RI, Retained intron; AF, Alternative first exon; AL, Alternative last exon) [116]. We run the command in its default "strict boundary" mode. We get as output:

```
gencode_v23_events_A3_strict.gtf
gencode_v23_events_A3_strict.ioe
gencode_v23_events_A5_strict.gtf
gencode_v23_events_A5_strict.ioe
gencode_v23_events_AF_strict.gtf
gencode_v23_events_AF_strict.ioe
gencode_v23_events_AL_strict.gtf
gencode_v23_events_AL_strict.ioe
gencode_v23_events_MX_strict.gtf
gencode_v23_events_MX_strict.ioe
gencode_v23_events_RI_strict.gtf
gencode_v23_events_RI_strict.ioe
gencode_v23_events_SE_strict.gtf
gencode_v23_events_SE_strict.ioe
```

To quantify an event type 'A3' for a given expression file we use the 'psiPerEvent' command (*Bash*):

```
>> python3 suppa.py psiPerEvent \
      -e transcript_TPM_suppa.tsv \
      -i gencode_v23_events_A3_strict.ioe \
      -o suppa_A3
```

The output we get is "suppa_A3.psi", which includes PSI values with a row for every splicing event and a column for every sample.

Survival models

We use the R *survival* package [126, 127] to perform Cox proportional hazards multivariate regression modeling. We start with a data table with four columns: Sample_Id (character); sample expression (numeric); sample purity (numeric); and subject age (integer). We quantile normalize the quantitative variables since the Cox method requires that inputs be normally distributed. We then merge in the survival data: EVENT (binary); TIME_TO_EVENT (integer); and SEX (categorical). For the model equation

$$h_{t,i} = h_0 \cdot \exp\left(\beta_{age} \cdot age_i + \beta_{purity} \cdot purity_i + \beta_{isoT} \cdot isoT_i + \beta_{geneT} \cdot geneT_i\right)$$

we use the command (*R*):

```
sa_output = try(coxph(Surv(TIME_TO_EVENT,EVENT) ~ \
      isoTPM + geneTPM + purity + Age + strata(SEX), data = SA_norm))
```

We assess our model using the a "try" function in case the model cannot be adequately assessed by the coxph function (this might be caused by, for example, insufficient event count, insufficient variable variance, or inability to find finite coefficient values). If we run this command using data for transcript GCDH-012 (ENST00000590472.5) the output is:

```
Call:
coxph(formula = Surv(TIME_TO_EVENT, EVENT) ~ Transcript_TPM +
    Gene_TPM + ESTIMATE + Age + strata(SEX), data = ranked_SA)

                  coef exp(coef) se(coef)      z        p
Transcript_TPM -0.7096    0.4918   0.2354 -3.015  0.00257
Gene_TPM        0.3058    1.3577   0.3204  0.954  0.33991
ESTIMATE        1.3188    3.7389   0.4119  3.201  0.00137
Age             0.7714    2.1628   0.1457  5.295 1.19e-07

Likelihood ratio test=48.74  on 4 df, p=6.616e-10
n= 350, number of events= 186
```

From this object we can extract our model coefficients ("coef") and their p-values ("p") as well as the model's initial and final log likelihood (LL) values (R):

```
tsum = summary(sa_output)
tsum[['loglik']][[2]]
```

which returns the final LL value of -785.0566. We compare this to the simpler model

$$h_{t,i} = h_0 \cdot \exp\left(\beta_{age} \cdot age_i + \beta_{purity} \cdot purity_i\right)$$

with the call (R):

```
sa_output = try(coxph(Surv(TIME_TO_EVENT,EVENT) ~ \
      purity + Age + strata(SEX), data = SA_norm))
```

and get this output:

```
Call:
coxph(formula = Surv(TIME_TO_EVENT, EVENT) ~ ESTIMATE + Age +
    strata(SEX), data = ranked_SA)

           coef exp(coef) se(coef)     z        p
ESTIMATE 1.2478    3.4827   0.3935 3.171  0.00152
Age      0.7833    2.1886   0.1467 5.339 9.35e-08

Likelihood ratio test=38.99  on 2 df, p=3.411e-09
n= 350, number of events= 186
```

Comparing Models

To compute the likelihood ratio test score for these models, we extract the final loglikelihood value from the simpler model (here, -789.9302) and the more complex model (here, -785.0566) and use the LRT equation to get the LRT score (here, 9.7472808). To get the Chi-squared p-value for this LRT score with its degrees of freedom (here, df=2 for the isoT and geneT variables) we use (*R*):

```
pchisq(q=LRT_score, df=2, lower.tail=FALSE)
```

which returns our p-value (here, p=0.00765). To correct our p-values for multiple tests we use the Benjamini-Hochberg method [128] via the base *R* function "p.adjust". For a results table results_dt we run (*R*):

```
pvals = results_dt$LRT_pval
padj = p,adjust(pvals, method="BH")
```

```
results_dt[,LRT_padj:=padj]
```

Results

Metadata & sample choices

Before running our survival models we explored the distributions of event and time-to-event values for two categorical variables to see if we needed to take them into account: subject sex (male or female); and sample type (primary tumor or metastatic).

Subject sex

Survival times for men and women for either event type are similarly distributed (**Figure 18**). We compared survival times for each event type between males and females using a two-sided Wilcoxon ranksum test and neither result was statistically significant. Nonetheless, large studies of melanoma patient survival based on US SEER data [129] and on five randomized trials from the EORTC Melanoma Group [130] find that independent of other factors patient sex has a significant effect on survival. Accordingly, we stratify our Coxph models by sex so that each sex has its own baseline hazard rate.

Sample type

Samples in the TCGA SKCM dataset are from either primary tumor or metastatic tissues. For melanoma patients with available age, purity, and survival data only four (out of 449) had both a metastatic and a primary tumor sample in the dataset. Of the samples with subject clinical data available, 350 of 451 (77.6%) are metastatic and 101 are primary (22.4%). Survival times for metastatic and primary samples for both event types are differently distributed (**Figure 19**). We

compared survival times for each event type between metastatic and primary samples using a two-sided Wilcoxon ranksum test and for both event types primary samples had significantly lower survival times (p<2.2E-16 for event '0'; p=3.9E-8 for event '1'). Accordingly to avoid having to further stratify our models we proceeded using only metastatic samples. For the 350 metastatic samples, 130 are female (37.1%), and there are 164 censored events (46.9%) and 186 deaths (53.1%).

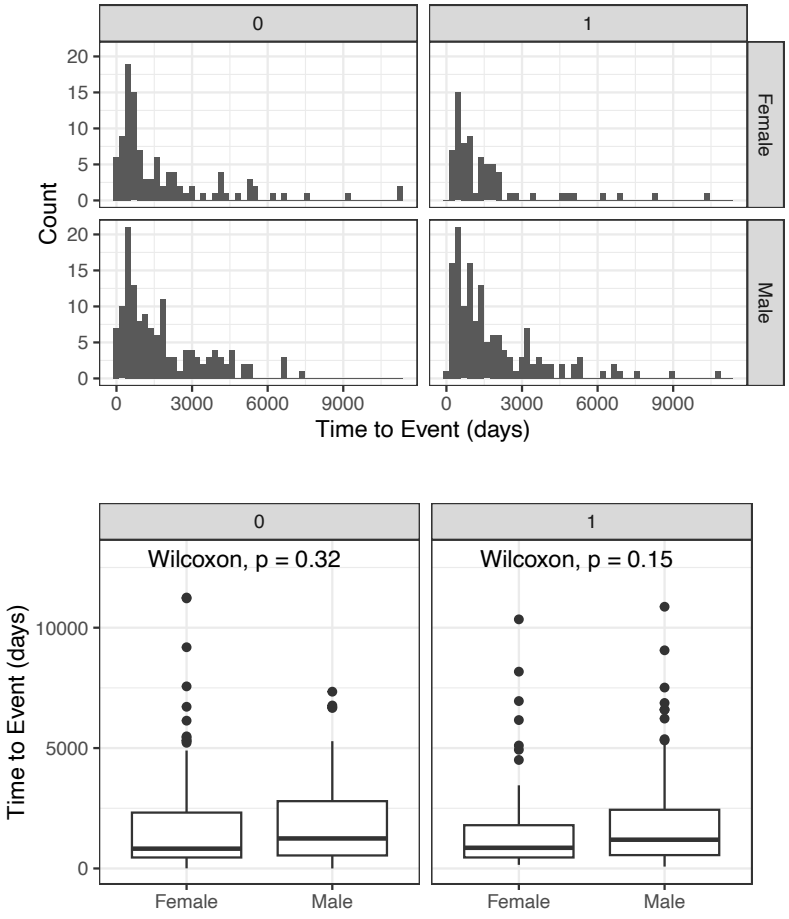


Figure 18. Male and female subjects have similar time-to-event values for both censor events and death events.
On the lefthand side, histograms show the distribution of event times for female (top row) versus male (bottom row) subjects for censor events ("0", lefthand column) and death events ("1", righthand column). On the righthand side, boxplots show distributions of event times for female versus male subjects for censor events ("0", lefthand panel) and for death events ("1", righthand

panel). Results for two-sided Wilcoxon rank sum tests comparing event times are shown in the plots.

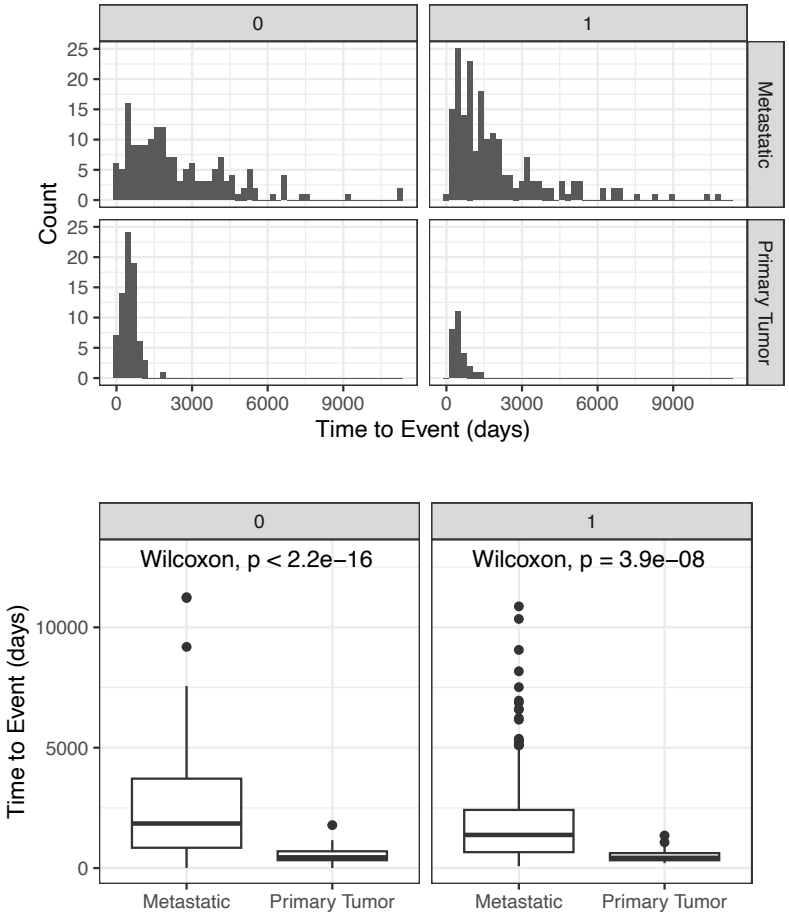


Figure 19. Subjects with primary tumor samples have significantly shorter time-to-event values than subjects without primary tumor samples.
On the lefthand side, histograms show the distribution of event times for metastatic (top row) versus primary (bottom row) tumor samples for censor events ("0", lefthand column) and death events ("1", righthand column). On the righthand side, boxplots show distributions of event times for metastatic versus primary tumor samples for censor events ("0", lefthand panel) and for death events ("1", righthand panel). Results for two-sided Wilcoxon rank sum tests comparing event times are shown in the plots.

Simple models for age and purity

Our Coxph models for age and purity show that both variables improve survival models. For our age model,

$$h_{t,i} = h_0 \cdot \exp\left(\beta_{age} \cdot age_i\right)$$

the LRT score is significant (Chi-squared test, p=1.45E-7, df=1) and shows that adding age improves the null model without age. The coefficient for age in this model is positive (0.699, p=4.36E-8), reflecting an increase in hazard with increased age. For our purity model,

$$h_{t,i} = h_0 \cdot \exp\left(\beta_{purity} \cdot purity_i\right)$$

the LRT score is significant (Chi-squared test, p=4.22E-4, df=1) and shows that adding purity improves the null model without purity. The coefficient for purity in this model is positive (0.478, p=7.41E-4), reflecting an increase in hazard associated with increased purity.

For our age+purity model,

$$h_{t,i} = h_0 \cdot \exp\left(\beta_{age} \cdot age_i + \beta_{purity} \cdot purity_i\right)$$

the LRT score is significant when compared to either the age model (Chi-squared test, p=1.21E-3, df=1) or the purity model (Chi-squared test, p=4.00E-7, df=1) showing that adding either purity to the age model or adding age to the purity model improves the simpler model. The coefficients for age and purity are similar to those seen in the simpler models (age coefficient=0.673, p=1.39E-7; purity coefficient=0.440, p=1.91E-3).

Survival models with gene expression variables

The results for our eleven multivariate gene expression Coxph model types are given in **Table 10.** For each model type we give: the number of survival events, the number of tests performed, the number of tests with LRT p-values significant at FDR<=20%; the number of significant

results with negative and with positive coefficients for the gene expression variable. For models with below-gene-level expression variables, in addition to individual model results we summarize results across below-gene-level models to give gene-level results. For example, if two IsoT+GeneT models for a gene are significant, that is counted as one significant model at the gene level. The FDR is not re-calculated for the gene-level summaries and so preserves the original multiple hypothesis correction from the (more numerous) below-gene-level models tested. The number of tests for a given model type reflects the number of suitable model inputs (e.g. whether for a given splicing event there were enough samples with a defined PSI value such that the number of events met the minimum for test power) as well as the number of those tests for which the Coxph function could solve the regression model.

Comparing model results coverage

We show the model type results counts and the counts for their unions in **Figure 20**. The lefthand bar plot shows the total number of significant test results by model type, in decreasing order from top to bottom. The righthand "upset" plot shows in the upper bar plot the number of significant test results found exclusively in each set of model types demarcated in the lower plot, in decreasing value from left to right.

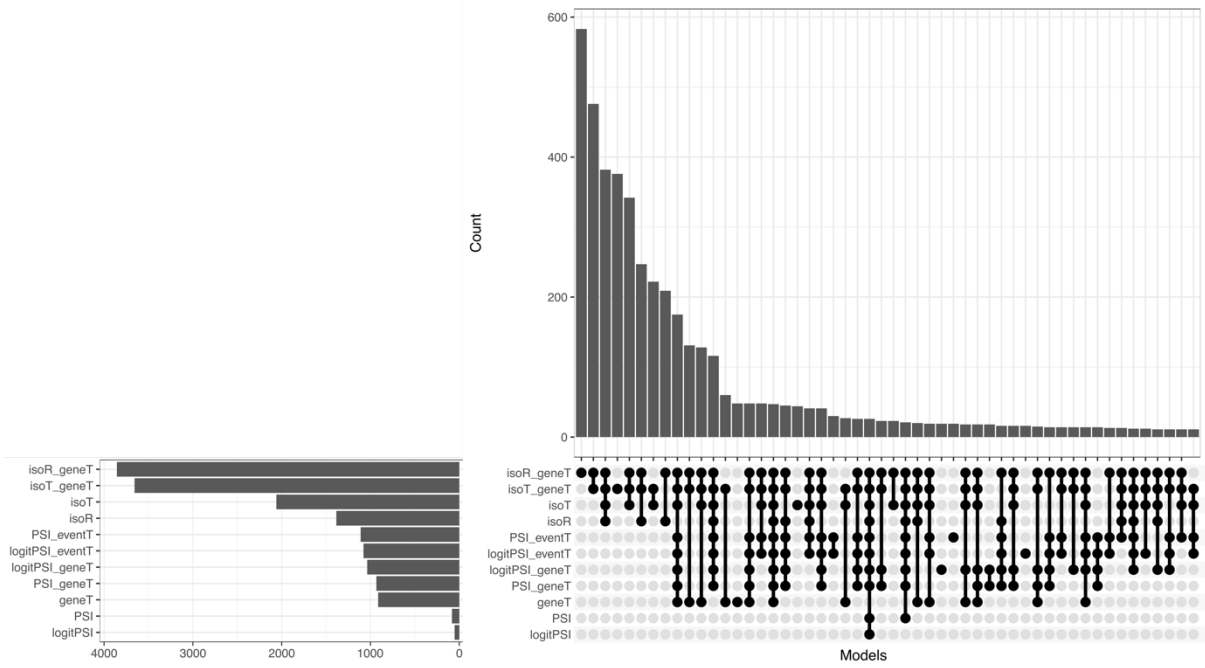


Figure 20. Results count for the eleven multivariate gene expression Coxph model types. The intersections were limited to the 50 with the most members (truncated the RHS of the top histogram etc.). Plot was built in *R* using *ggplot2* [131] and *ggupset* [132] packages and truncated using Inkscape.

Table 10. Output statistics for the eleven multivariate gene expression Coxph models.

Model	Events	Gene level				Below gene level			
		Tests	FDR 20%	Neg Coef	Pos Coef	Tests	FDR 20%	Neg Coef	Pos Coef
GeneT	186	14382	914 (6.4%)	600 (65.6%)	314 (34.4%)	---	---	---	---
IsoT	186	14376	2060 (14.3%)	1678 (81.5%)	403 (19.6%)	129911	2763 (2.1%)	2298 (83.2%)	465 (16.8%)
IsoR	37-186	14376	1385 (9.6%)	1218 (87.9%)	220 (15.9%)	129906	1603 (1.2%)	1382 (86.2%)	221 (13.8%)
PSI	30-186	13331	83 (0.6%)	55 (66.3%)	34 (41%)	135732	103 (0.1%)	63 (61.2%)	40 (38.8%)
logitPSI	30-186	13331	53 (0.4%)	34 (64.2%)	21 (39.6%)	135732	63 (0%)	38 (60.3%)	25 (39.7%)
IsoT + GeneT	186	14376	3657 (25.4%)	3084 (84.3%)	1552 (42.4%)	129911	9074 (7%)	6072 (66.9%)	3002 (33.1%)
IsoR + GeneT	41-186	14374	3856 (26.8%)	3374 (87.5%)	1644 (42.6%)	129902	9387 (7.2%)	6329 (67.4%)	3058 (32.6%)
PSI + GeneT	40-186	13302	934 (7%)	678 (72.6%)	450 (48.2%)	133189	1870 (1.4%)	1154 (61.7%)	716 (38.3%)
logitPSI + GeneT	40-186	13302	1037 (7.8%)	759 (73.2%)	511 (49.3%)	133189	2184 (1.6%)	1340 (61.4%)	844 (38.6%)
PSI + eventT	40-186	13302	1111 (8.4%)	777 (69.9%)	529 (47.6%)	133189	1870 (1.4%)	1150 (61.5%)	720 (38.5%)
logitPSI + eventT	40-186	13302	1079 (8.1%)	763 (70.7%)	513 (47.5%)	133189	1860 (1.4%)	1141 (61.3%)	719 (38.7%)

Discussion

Single expression level variable models

Out of the five single expression-variable models, the GeneT, IsoT, and IsoR models had many more significant results at the gene level (914, 2060, and 1385, respectively) than the PSI and logitPSI models (83 and 53, respectively). In addition, the IsoT and IsoR models had more gene-level results than the GeneT models, despite the fact that the IsoT and IsoR models involved at

least nine times as many tests (129911 and 129906 versus 14382) and thus had a greater multiple hypothesis correction penalty.

Multilevel expression variable models

Out of the six multilevel expression variable model types, IsoT+GeneT and IsoR+GeneT had many more significant results at the gene level (3657 and 3856 results, respectively) than the PSI+GeneT, logitPSI+GeneT, PSI+EventT, and logitPsi+EventT models (934, 1037, 1111, and 1079 results, respectively).

Comparing model coverage

In **Figure 20** we see that the isoR_geneT model type had the most gene-level results not found by other models (n=583), followed by the combination of isoR_geneT and isoT_geneT model types (n=476). Importantly, only 48 out of the total 4853 genes (0.99%) with significant results were found exclusively in the geneT model type. Similarly only 151 out of the total 4853 genes (3.11%) with significant results were found in the union of the six models which include splicing event (PSI, logitPSI, or eventT) variables but not in the union of the five other models.

The isoR+geneT and isoT+geneT model types are clearly the most predictive. Together these two model types capture 4603 of the 4853 (94.8%) significant gene-level identifications (3856 and 3657 individually), and adding to that set the isoT and isoR model results would only add 50 genes (1.1%) for a total of 95.9% of all gene identifications. There are only 50 genes (1.1%) that are found by the geneT models but not the isoR+geneT and isoT+geneT model types.

The Venn diagrams in **Figure 22** and **Figure 21** compare model results. The comparison of geneT versus geneT+below-gene-level models (**Figure 22**, lefthand column) and below-gene-

level models versus below-gene-level+geneT models (**Figure 22**, righthand column) shows that the two-level isoR+geneT and isoT+geneT models capture nearly all of the gene identifications found by the single-variable models using either metric individually. In contrast, PSI and logitPSI models capture very few identifications not found in the (logit)PSI+geneT models. The comparison and (logit)PSI versus (logit)PSI+eventT models (**Figure 21**) shows a similar result for models using (logitPSI) or eventT or both. We also see in **Figure 20** that (logit)PSI+eventT model types slightly outperform (logit)PSI+geneT model types.

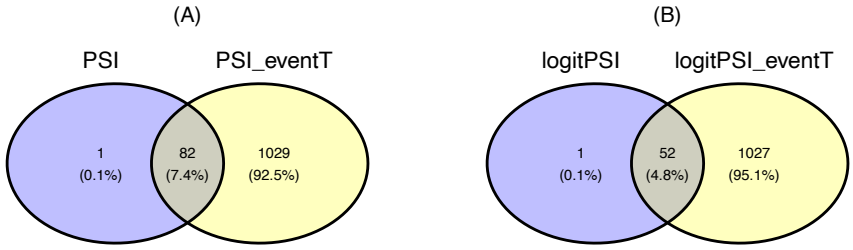


Figure 21. Venn diagrams showing significant gene-level results for (A) PSI versus PSI+eventT models and (B) logitPSI versus logitPSI+eventT models.
Plot was built in *R* using *ggplot2* [131] and *ggvenn* [133] packages.

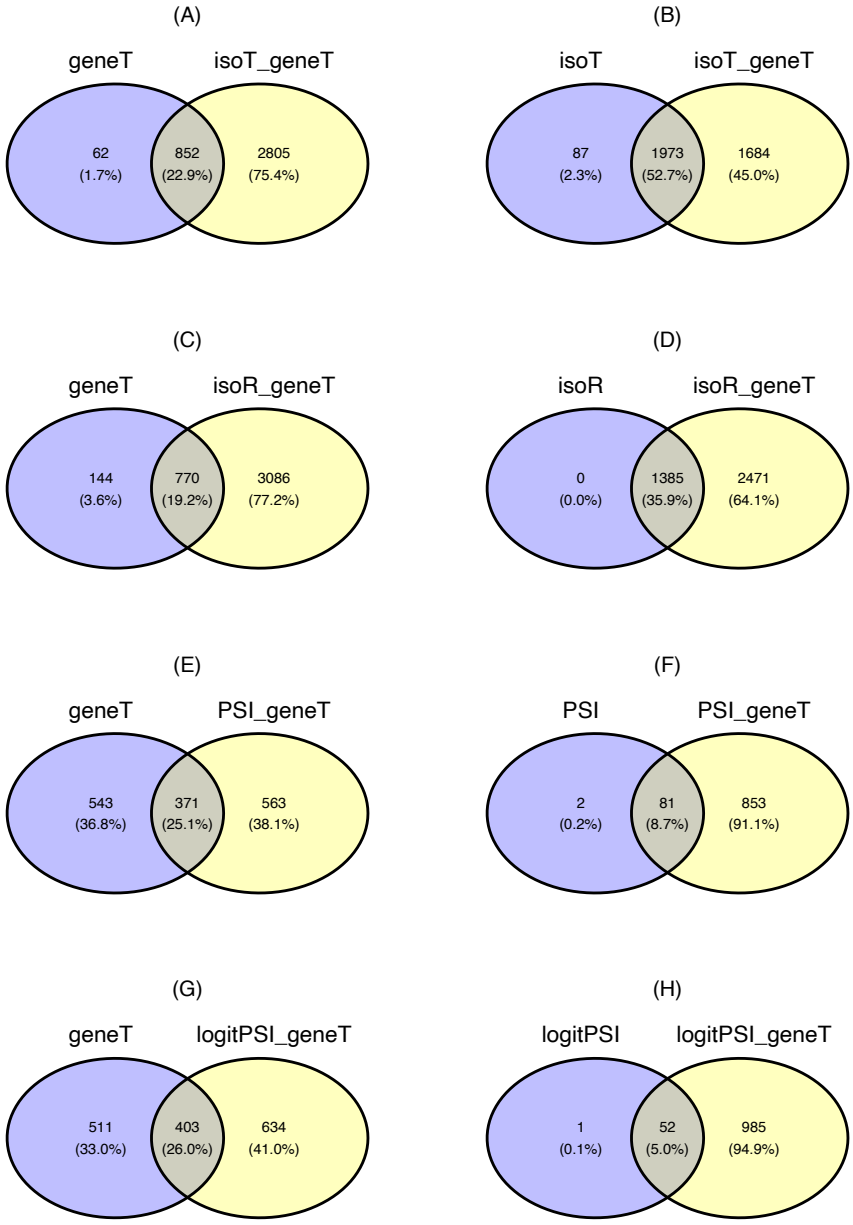


Figure 22. Venn diagrams showing significant gene-level results for single- versus multi-level Coxph models.
The lefthand column compares geneT models to geneT+below-gene-level models. The righthand column compares below-gene-level models to geneT+below-gene-level models. Plot was built in *R* using *ggplot2* [131] and *ggvenn* [133] packages.

Case study: GCDH splice variants

The association between glutaryl-CoA dehydrogenase (GCDH) splicing isoform expression and patient survival provides a good case study to illustrate the utility of multilevel expression models. In an earlier version of this project we aimed to identify protein-coding genes whose gene-level expression was not associated with survival in melanoma but whose isoform-level expression was. We focused on genes for which one isoform was positively associated with survival and another isoform was negatively associated with survival. We thought this would provide more evidence for the gene's survival relevance than a single isoform result and the structural differences between the two isoforms might provide some insight into the role this gene might play in melanoma survival (e.g. a canonical versus a truncated or "short" isoform).

We highlighted two groups of genes from our survival analysis. First, those sixteen with one significant hazardous isoform, and one significant helpful isoform, of which four had hazardous canonicals (APOOL, BABAM1, GCDH, and TMEM228). Second, those six involved in glutamate metabolism of which two had hazardous short isoforms (HAL, FPGS), two had helpful short isoforms (GLS, ALDH5A1), and two had helpful canonical isoforms (GAD1, GCLM). Our experimental collaborators tested the six glutamate genes, two of the bivalent genes (BABAM1, GCDH), and three other hazardous genes, two with hazardous short isoforms (GPS1, OTUD6B) and one with a hazardous canonical isoform (OTUD6B). Of these ten, the only gene whose si-RNA reduced melanoma growth (A375) was GCDH. Follow-up experimental work found the that si-RNA treatment against GCDH reduced cell viability for two other melanoma cell lines (UACC908, 1205LU) but not for liver (SK-HEP1, PLC), breast (SKBR3, MCF7) or prostate (DU145, PC3) cell lines [134]. Our collaborators found that GCDH knockdown increases the supply of glutaryl-CoA, which facilitates glutarylation of NRF2, increasing the

stability and DNA binding activity of NRF2 and leading to upregulation of cell death pathways. In mouse models, in vivo GCDH knockdown inhibited melanoma growth. Our collaborators are currently developing GCDH-targeting small molecules for potential use in human cancer treatment.

Table 11 provides results from the current analysis for GCDH models (the earlier analysis involved only geneT and isoT+geneT models). We can see that only models including below-gene-level variables identified GCDH as significantly affecting survival, and only the isoT+geneT model type identified both a hazardous isoform (GCDH-002) and a helpful isoform (GCDH-012).

Table 11. Select fitted Coxph model outputs for GCDH gene expression variables. There were 186 events for each model.

Model	Id Type	Feature Name	Coxph Coef	Coxph Pr	LRT padj
Passing FDR<=20%					
isoT	Transcript Id	GCDH-012	-0.62	3.70E-03	1.90E-01
isoT+ geneT	Transcript Id	GCDH-012	-0.71	2.57E-03	1.68E-01
isoT + geneT	Transcript Id	GCDH-002	1.36	5.97E-04	1.19E-01
isoR + geneT	Transcript Id	GCDH-012	-0.74	3.40E-03	1.81E-01
Not passing FDR<=20%					
geneT	Gene Id	GCDH	0.14	6.08E-01	8.50E-01
isoT	Transcript Id	GCDH-002	0.58	6.22E-02	4.29E-01
isoR	Transcript Id	GCDH-012	-0.68	4.51E-03	2.18E-01
isoR	Transcript Id	GCDH-002	0.30	3.10E-02	3.60E-01
isoR + geneT	Transcript Id	GCDH-002	0.33	2.87E-02	3.57E-01
PSI	Event Id	AL: GDCH-012, GCDH-002	-0.62	4.46E-03	3.64E-01
logitPSI	Event Id	AL: GDCH-012, GCDH-002	-0.61	4.46E-03	3.63E-01
PSI + eventT	Event Id	AL: GDCH-012, GCDH-002	-0.63	9.20E-03	2.26E-01
logitPSI + eventT	Event Id	AL: GDCH-012, GCDH-002	-0.63	9.20E-03	2.27E-01

Conclusion

We have shown for the TCGA metastatic melanoma dataset that adding below-gene-level RNA-seq metrics to multivariate Cox models greatly increases the detected associations. In addition we have demonstrated the utility of analyzing below-gene-level expression using the isoR and eventT metrics. These metrics show promise for helping us better capture splicing dynamics in our multivariate survival models.

Acknowledgements

This work utilized the computational resources of the NIH HPC Biowulf cluster (https://hpc.nih.gov). I would like to thank members of the Ronai Lab and our other collaborators on the GCDH project [134] for their many significant contributions.

Chapter 4: Heteroformity and Entropy as Measures of Splicing Diversity

Introduction

The differential expression of spliced isoforms in human tissues varies by tissue type and gene [135] and may play important roles in cancer biology [13, 14]. The extent to which these splicing differences result from functional regulatory [7, 8, 9] versus stochastic mechanisms [10, 11, 12] is a subject of lively debate. Our aim in this chapter is not to weigh in on these issues but rather to explore how diversity measures adapted from population genetics and population ecology might help us measure and compare splicing diversity. We hope the discussion here will provide some useful metrics and ideas for discussing questions like "How much alternative splicing is there in stomach as opposed to testes cells?" or "How much does alternative splicing increase under this new condition?"

Metrics

We focus on three related diversity indices: count, Shannon-Weaver entropy, and the Gini-Simpson index. This section is based largely on Jost's treatment of entropy and diversity in population ecology [136] and Nei's treatment of diversity in population genetics [137].

Indices

The first index, species richness (species count), is given in terms of abundance:

$$^{C}I = n$$

where a region has *n* species. In our splicing case, we take ^{C}I where a gene has *n* isoforms.

The second index, Shannon-Weaver entropy, is given in terms of information:

$$^{s}I = -\sum_{i=1}^{n_j} p_i \log p_i$$

where p_i is the relative abundance (count over total) of species *i* in a region. In our case, we take ^{s}I where for a gene with *n* isoforms, p_i is the relative abundance of isoform *i*, given by $p_i = t_i / \sum_{j=1}^{n} t_j$, for isoforms *i*, $i = (1,2,\ldots,n)$, with transcript counts t_i, and where $p_i \log p_i$ is taken to be equal to zero when $p_i = 0$.

The third index is the Gini-Simpson index, given in terms of probability:

$$^{h}I = 1 - \sum_{i=1}^{n} {p_i}^2$$

where p_i is the relative abundance (count over total) of species *i* in a region. For Simpson (1949), the term $\sum {p_i}^2$ (above) "can be simply interpreted as the probability that two individuals chosen at random and independently from the population will be found to belong to the same group [within that population]" [138]. For Nei, $1 - \sum {p_i}^2$ gives gene diversity for a population [137]; under panmixis, the sum might be called homozygosity and 1 minus the sum heterozygosity. In our case, we take ^{h}I where p_i is the relative abundance of isoform *i* (and so forth), as with Shannon-Weaver entropy. Thus, we interpret $1 - \sum {p_i}^2$ as the probability that for a given gene two independent transcript selections correspond to different isoforms. We call this measure *heteroformity* (it is the gene:isoform analogue of heterozygosity).

Diversities

The three indices are in fact closely related to different orders of the Hill equation [139, 136]:

$$^{q}D = \left(\sum_{i=1}^{n} {p_i}^{q}\right)^{1/(1-q)}$$

These Hill diversities are given in common units, the effective number of species (for us, the effective number of isoforms), and provide equivalence classes: the number of equally abundant species (isoforms) resulting in the same diversity (heteroformity). Thus, whereas it is not clear how to quantitatively compare indices (e.g. information units of entropy and probability units of heteroformity), Hill diversities allow a direct, intuitive comparison. The major difference between the three measures is how much weight they give to rarer or more common types. To visualize how heteroformity and entropy values are related we plot their respective maximum values across a range of splicing scenarios holding either the count of the less frequent ("minor") isoforms constant (**Figure 23**, top) or the frequency of the more frequent ("major") isoform constant (**Figure 23**, bottom).

The 0th-order Hill number is species richness (here the index is equivalent to the diversity),

$$^{0}D = \left(\sum_{i=1}^{n} {p_i}^{0}\right)^{1/(1-0)} = n = {}^{c}I$$

The 1st-order Hill number is undefined, but its limit exists and equals the exponential of Shannon-Weaver entropy,

$$^{1}D = \exp\left(-\sum_{i=1}^{n} p_i \ln p_i\right) = \exp({}^{s}I)$$

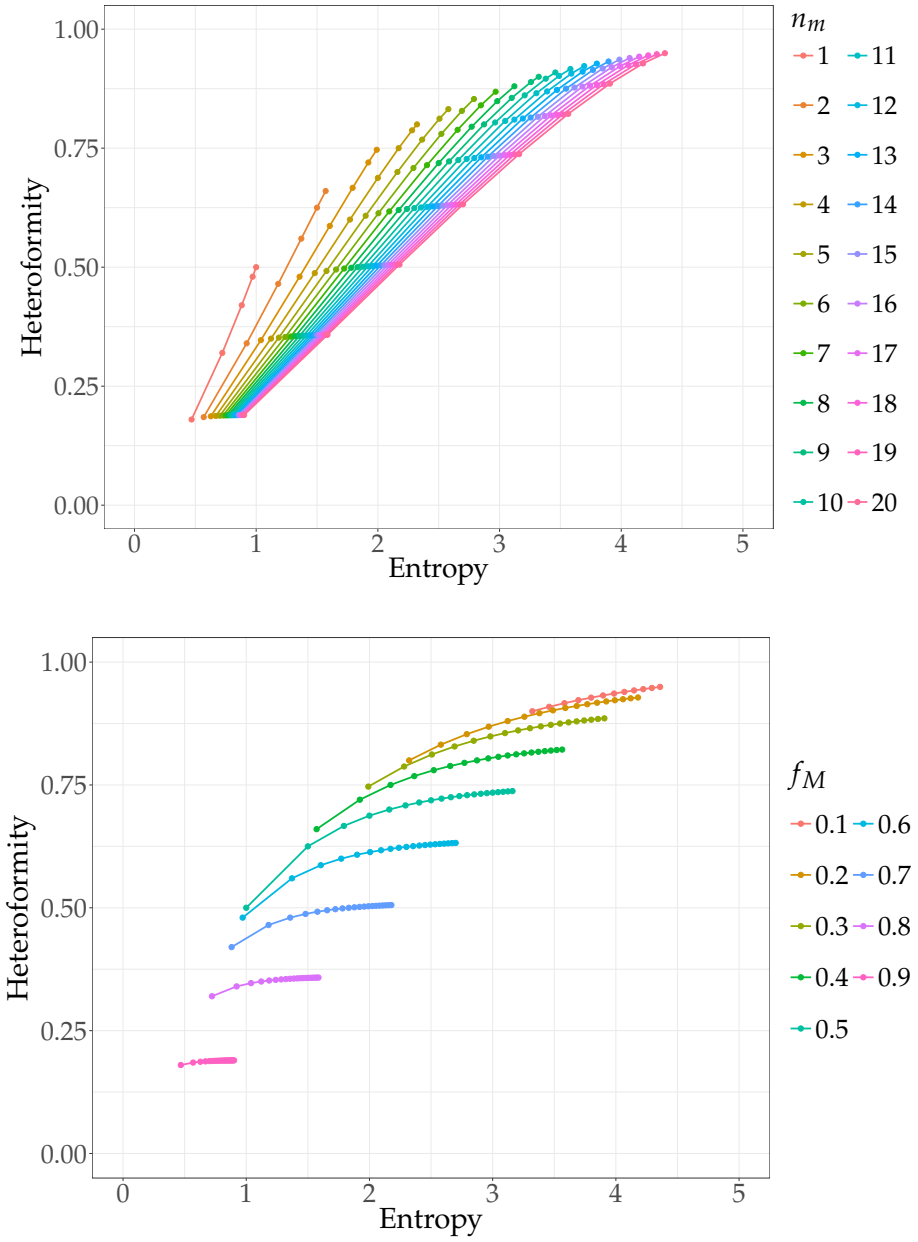


Figure 23. Maximum heteroformity and entropy values for genes with fixed major isoform frequencies or fixed minor isoform counts.
For a gene with major isoform frequency f_M, each of n_m minor isoforms has frequency $f_m = (1 - f_M)/n_m$ where $10 \times (1 - f_M) \leq n_m \leq 20$. The lower bound on n_m is such that $f_M \leq f_m$; the upper bound on n_m is arbitrary. The clines each have a constant n_m (top) or a constant f_M (bottom). Plot made using *knitr* [140]

The 2nd-order Hill number equals $1/(1 - {}^{h}I)$, or the inverse of the probability of two independent transcript selections corresponding to the same isoform,

$$^{2}D = 1/\left(\sum_{i=1}^{n} {p_i}^2\right) = 1/(1 - {}^{h}I)$$

Comparative Diversity

We can also apply measures to compare diversity between populations. Following Nei [137], we have gene identity J_i and gene diversity H_i for population *i*:

$$J_i = \sum_k {f_{ik}}^2$$

$$H_i = 1 - \sum_k {f_{ik}}^2$$

where $0 < J_i \leq 1$ and $0 \leq H_i < 1$. For gene identity between subpopulations J_{ij} and gene diversity between subpopulations H_{ij}:

$$J_{ij} = \sum_k f_{ik} f_{jk}$$

$$H_{ij} = 1 - \sum_k f_{ik} f_{jk}$$

Here we address the probability of two individuals each picked at random and independently from separate groups are of the same type. In our splicing case we could refer to the heteroformity analogue of H_{ij} as *comparative heteroformity* as it compares two groups by drawing a sample from each.

Nei [137] defines gene diversity between populations as

$$D_{ij} = H_{ij} - \left(H_i + H_j\right)/2$$

$$= \left(J_i + J_j\right)/2 - J_{ij}$$

and he goes on to show that

$$D_{ij} = \sum_k \left(f_{ik} - f_{jk}\right)^2/2$$

which he describes elsewhere as a measure of genetic distance between two populations. In the case of splicing we could refer to the heteroformity analogue of Nei's D_{ij} as *differential heteroformity*.

In the case of entropy, we can use comparative information measures. Jensen-Shannon Divergence provides a symmetrical measure of entropy distance between two populations P and Q where, using log base 2, $0 \leq JSD(P \parallel Q) \leq 1$:

$$JSD(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M)$$

where $M = (P + Q)/2$. This has the advantage over Kullback-Leibler divergence of being defined when $Q(i) = 0$ does not imply $P(i) = 0$. (Kullback-Leibler divergence is defined only if for all i, $Q(i) = 0$ implies $P(i) = 0$ (absolute continuity); when $P(i) = 0$, the ith term contributes zero because $\lim_{x \to 0}(x \log x) = 0$:

$$D_{KL}(P \parallel Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}$$

$$= \sum_i P(i) log \frac{Q(i)}{P(i)}$$

Methods

We analyzed two datasets. First, we computed heteroformity and entropy values for genes expressed in GTEx v8 normal human tissue samples [47]. We used this data to explore differential splicing of single genes and across all genes between different tissues. Second, our collaborators at Carnegie Mellon University subsampled raw RNA-seq data for five TCGA lung adenocarcinoma samples and provided us with processed isoform-level expression data for samples containing 10%, 20%, 50%, and 100% of their original raw reads. We used this data to explore how the accuracy of heteroformity and entropy measures for subsampled isoform results when compared to the full sample isoform results are affected by the diversity and expression levels of the genes in question.

Results & Discussion

Visualizing heteroformity

In this section we focus on heteroformity rather than on entropy because we find the bounds of heteroformity [0,1) and heteroformity's probabilistic interpretation more intuitive than the same for entropy. For a given gene, we can compare its heteroformity across tissues (**Figure 24**, bottom) much like we would compare its abundance across tissues (**Figure 24**, top). In **Figure 24** we compare these values for the gene NR3C1 (chosen because it has widely varying values for these measures across tissues) across 30 GTEx tissues. We can see that in this case the across-tissue patterns of these metrics for NR3C1 differ greatly from one another. For example, we could say that for this gene expression is relatively high in brain subtissues compared to other tissues, whereas for this gene heteroformity is relatively low for most of the brain subtissues compared to other tissues.

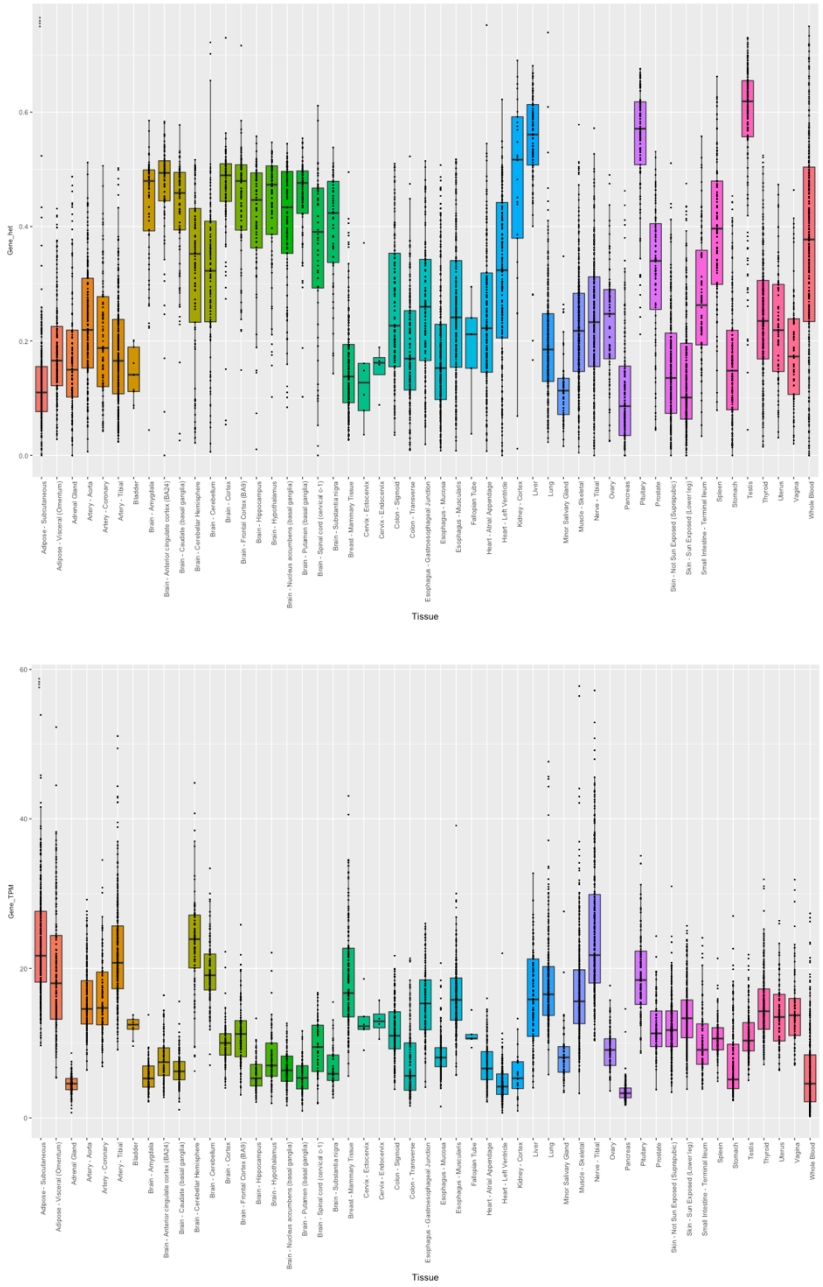


Figure 24. Distributions for gene NR3C1 of gene heteroformity (top) and gene TPM (bottom) for 30 GTEx tissues.
Boxplots are bounded by the first and third quartiles, with the mean value in the middle. Whiskers represent 1.5*IQR (interquartile range) on either end of boxes. Individual points indicate single sample values. Plots made using *ggplot2* [131].

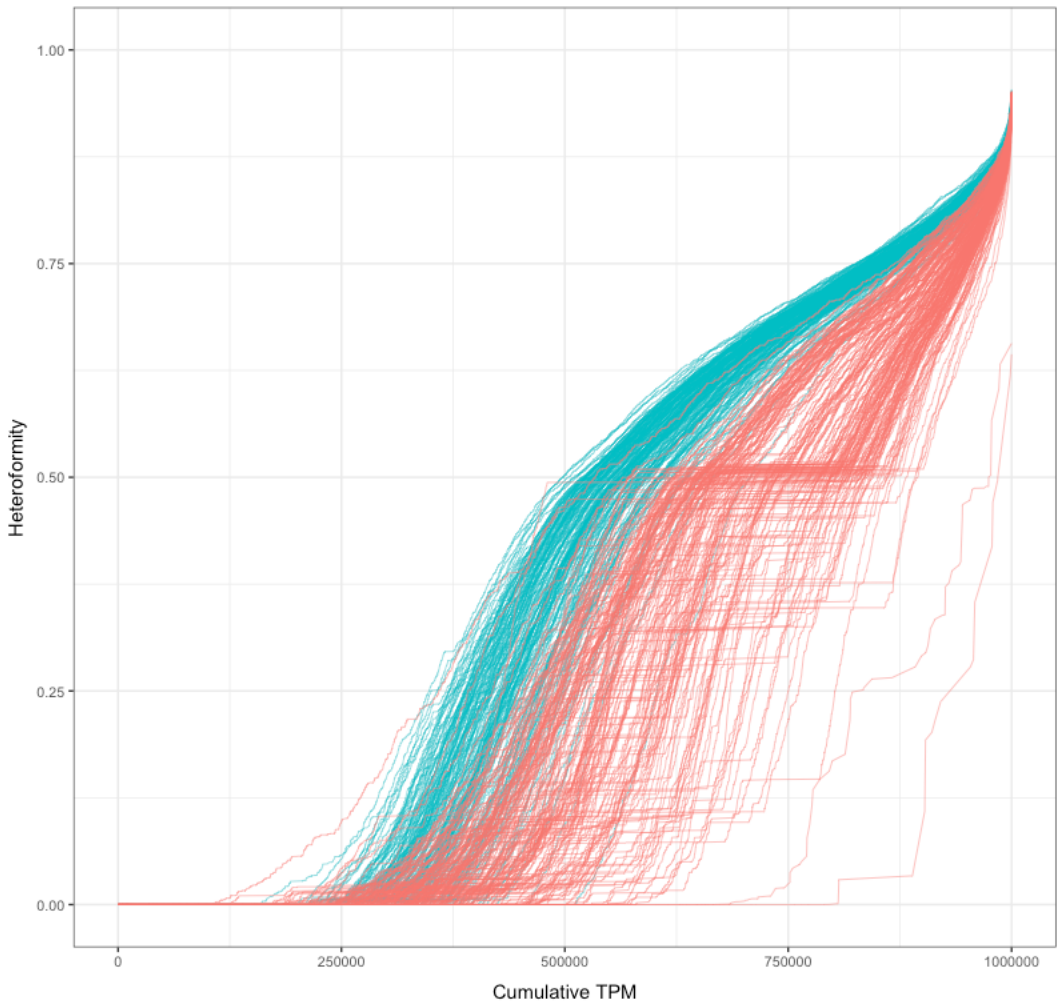


Figure 25. Comparison of sample heteroformity values as a function of cumulative TPM for GTEx testis (n=322, blue) and GTEx stomach (n=324, red) datasets.
Plot made using *ggplot2* [131].

We can also express splicing heteroformity across genes for a sample as the area under the curve (AUC) for the sample's cumulative splicing curve. In this across-gene approach we weight a gene's heteroformity by that gene's relative abundance. In **Figure 25** we show the cumulative heteroformity curves for GTEx testes (n=322) and stomach (n=324) samples. We can see that testes samples tend to have higher and less variable cumulative heteroformity values than stomach samples. We visualize the tissue-level distributions of sample cumulative

heteroformity values for 30 GTEx tissues in **Figure 27**. Here we can see for example that cumulative heteroformity is lowest in GTEx heart samples and highest in GTEx ovary samples. We visualize tissue-level average cumulative heteroformity curves (averaged by gene across samples within a tissue for 30 GTEx tissues in **Figure 26**.

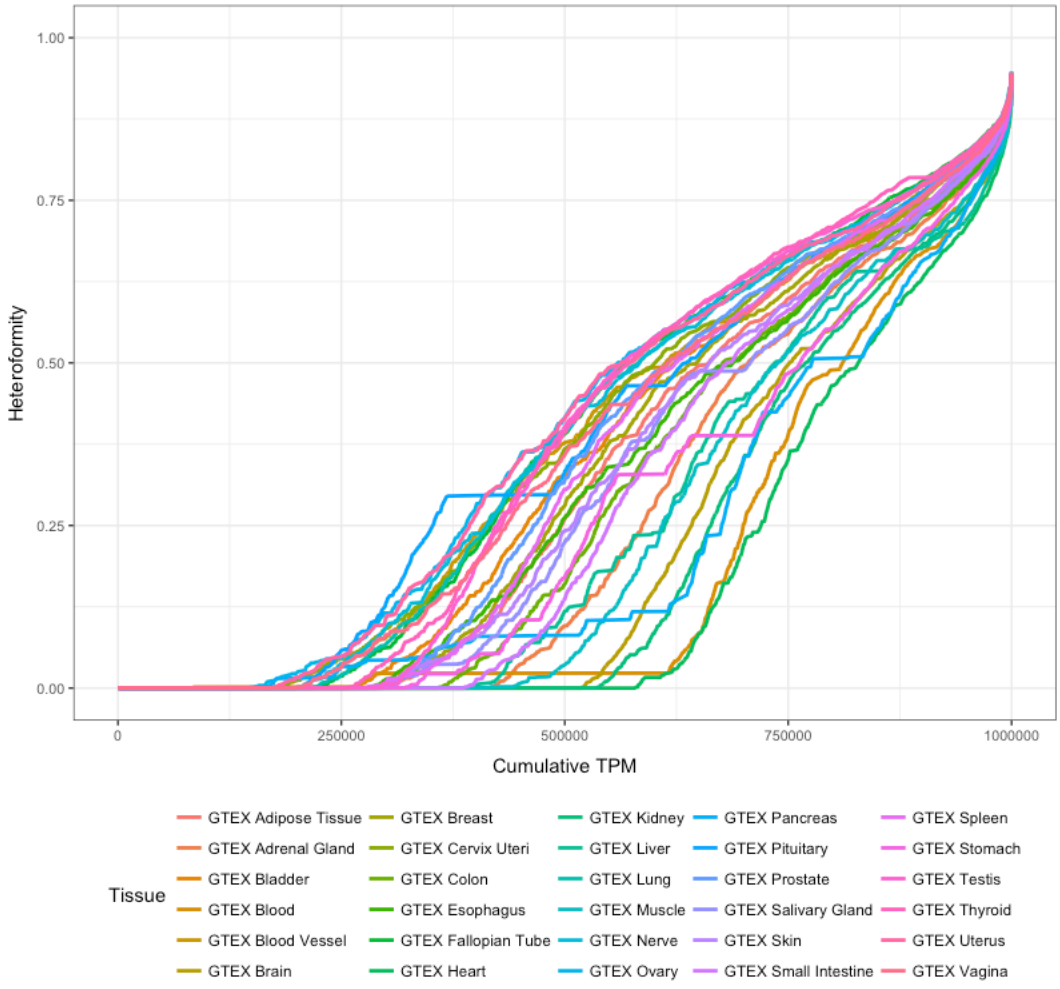


Figure 26. Tissue-level average (gene averages across samples within a tissue) heteroformity curves for 30 GTEx tissues.
Plot made using *ggplot2* [131].

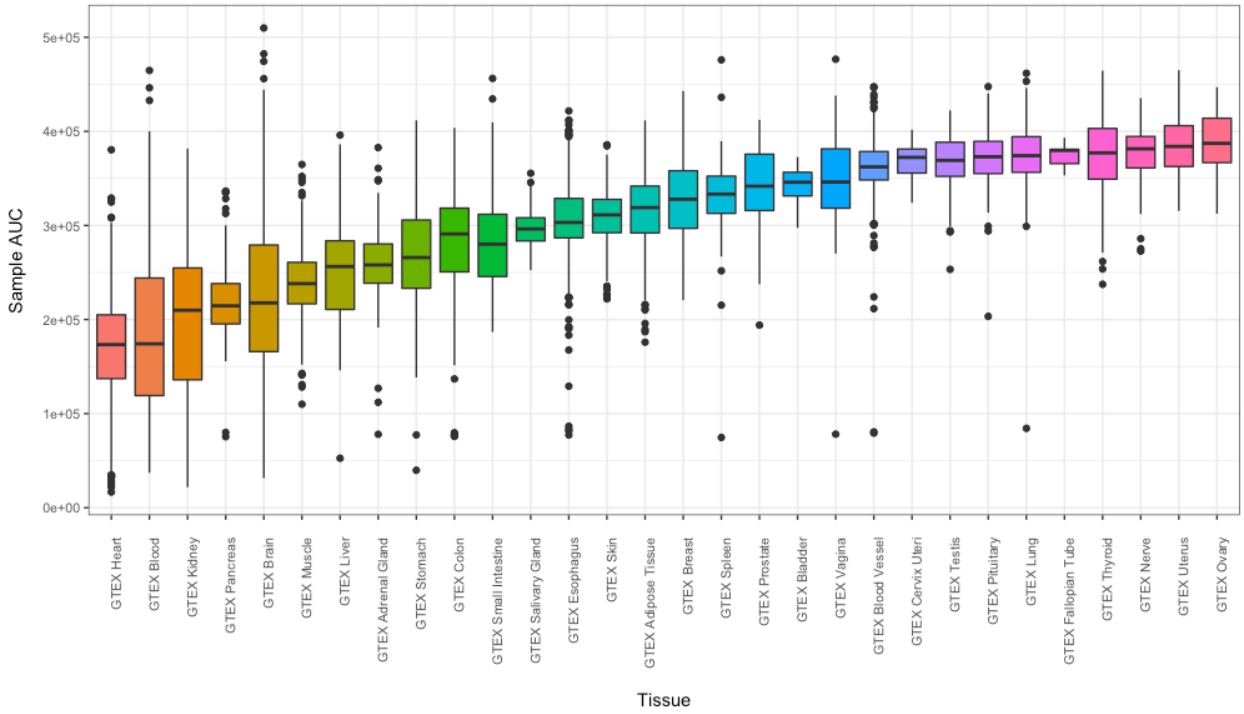


Figure 27. Sample heteroformity curve AUCs for 30 GTEx tissues.
Plot made using *ggplot2* [131].

Heteroformity and entropy errors and subsampling

In **Figure 29** we compare the relative sensitivity of heteroformity and entropy to subsampling at the gene level as a function of gene abundance and gene diversity in our TCGA lung cancer data. (The results are similar when using either heteroformity or entropy as the diversity measure.) For each gene abundance quartile and gene diversity quartile we compare for each gene the across-sample average error of heteroformity and entropy measures at the 10% subsampling level. We measure error in terms of the subsampled data value's difference from full sample data value using the coefficient of variation,

$$\mathrm{CV} = \frac{\sigma}{\mu}$$

where σ is the standard deviation and μ is the arithmetic mean of the differences between the subsampled values and the full values. In **Figure 29** we plot the difference for each gene between the CV for heteroformity and the CV for entropy. Accordingly, when the distribution is shifted left of center the entropy measure errors are relatively greater and when the distribution is shifted right of center the heteroformity measure errors are relatively greater. In general, we see that heteroformity errors are greater for genes with lower diversity whereas entropy errors are greater for genes with higher diversity. We also observe that the relative error distributions become more peaked in higher gene abundance quartiles. As we might expect, the distributions of relative errors across diversity quantiles become taller and narrower as gene abundance increases.

In **Figure 28** we compare relative measure errors between heteroformity and entropy for as a function of the level of RNA-seq read subsampling (rows, for 10%, 20%, and 50% subsampling) and either the number of gene transcripts expressed in the full RNA-seq data (columns, top plot) or gene heteroformity in the full RNA-seq data (columns, bottom pot). Across both gene transcript count quintiles and gene heteroformity quintiles, as these values increase heteroformity error decreases and entropy error increases. As we might expect, the distributions of relative errors become taller and narrower as the subsampling level increases.

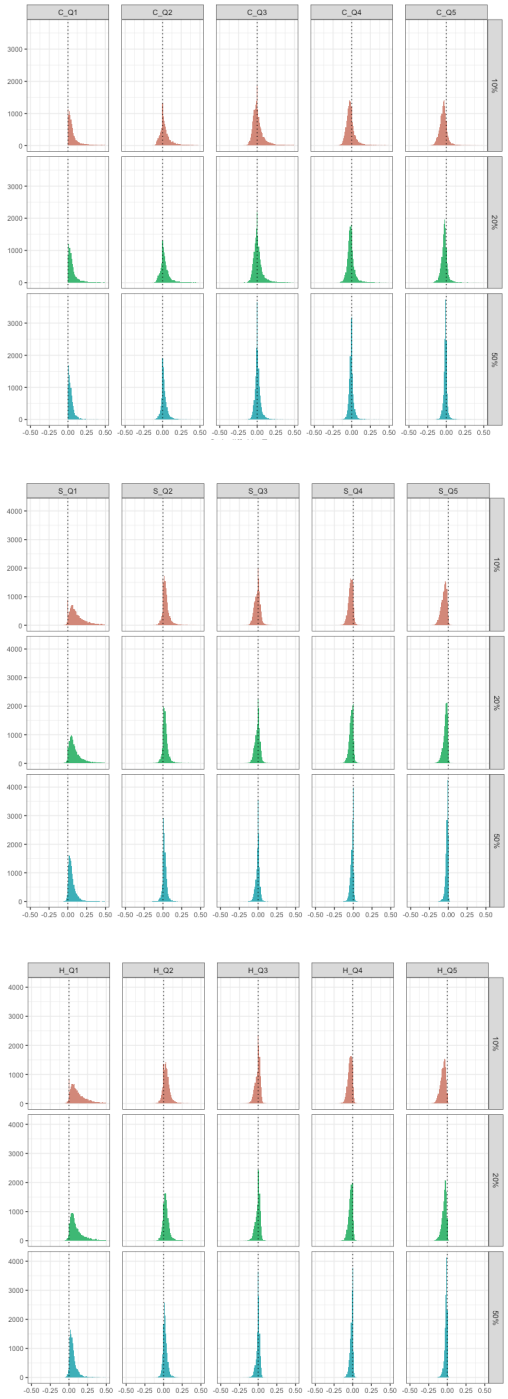


Figure 28. Gene-level counts (y-axis) for across-sample average differences between coefficients of variation for heteroformity and entropy (x-axis) as a function of the level of RNA-seq read subsampling (rows) and the number of gene transcripts expressed in the full RNA-seq data (columns, top), gene entropy in the full RNA-seq data (columns, middle), or gene heteroformity in the full RNA-seq data (columns, bottom) for our lung cancer samples. Plots made using *ggplot2* [131].

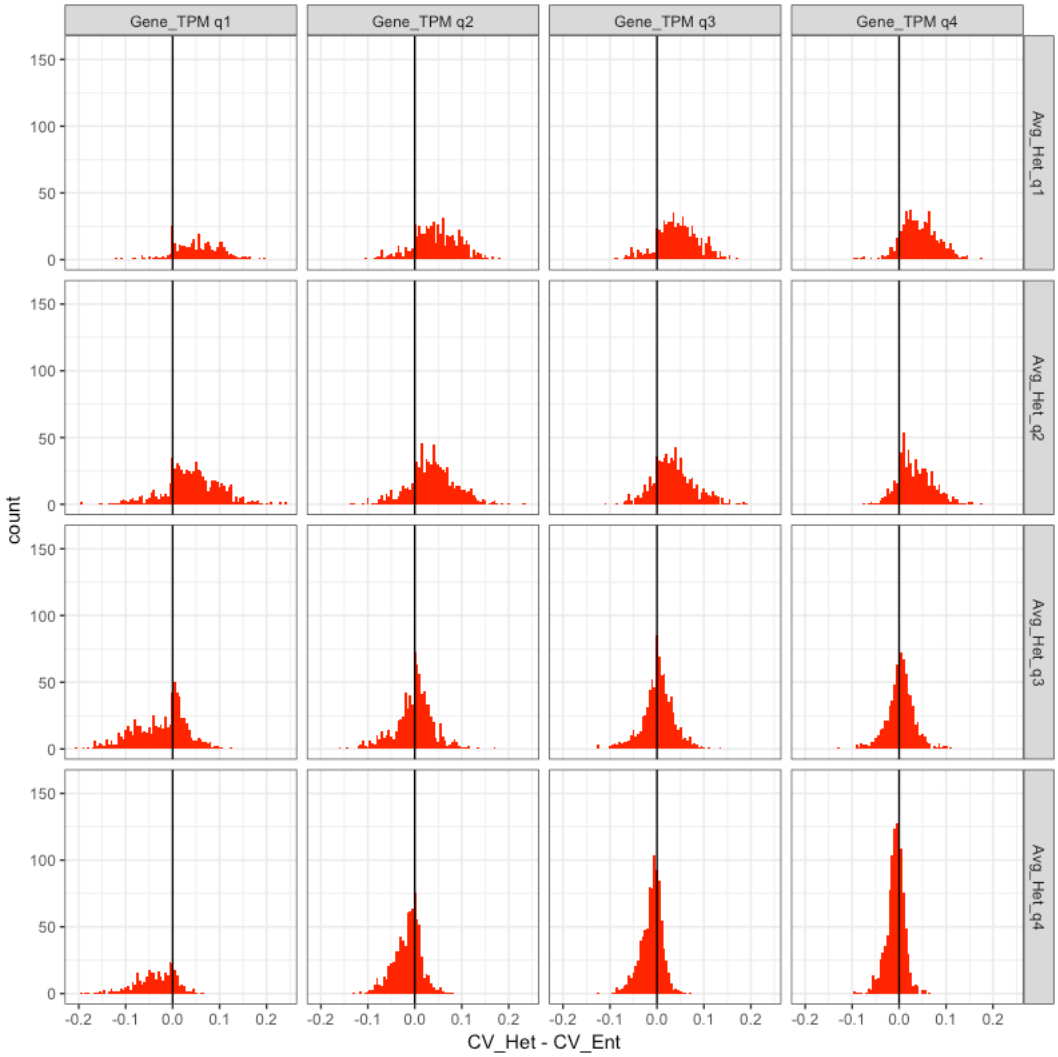


Figure 29. Average differences between gene heteroformity coefficients of variation and gene entropy coefficients of variation for subsampled (20%) LUAD RNA-seq data as a function of gene TPM quartile (columns) and average heteroformity quartile (rows).
Positive values (to the right of the black line) indicate more variation in heteroformity estimates than in entropy estimates. Negative values (to the left of the black line) indicate more variation in entropy estimates than in heteroformity estimates. Distributions using average entropy quartiles for the rows are similar (not shown). Plot made using *ggplot2* [131].

Conclusion

In our brief exposition we have seen that splicing diversity can be compared visually across samples or tissues for individual genes or for the entire transcriptome using gene heteroformity or cumulative splicing heteroformity.

We have also seen that whereas both heteroformity and entropy can be used to measure splicing diversity, their relative accuracy varies depending on the gene being measured. Gene heteroformity is less sensitive to sampling error (represented via our subsampling process) for genes with high expressed transcript counts or high splicing diversities. Conversely, splicing entropy is less sensitive to sampling error for genes with low expressed transcript counts or low splicing diversities. Errors for both measures decrease as gene abundance increases.

Chapter 5: Discussion

The projects discussed here have demonstrated the breadth of applications of next-generation sequencing data and novel computational techniques in addressing cancer research questions. Future research will benefit from many large-scale projects that continue to expand available "big data" for computational analyses [141]. In particular, more comprehensive long-read RNA-seq datasets will help researchers better refine transcript-level analyses and better understand the dynamics and impacts of mRNA splicing [142].

Sex bias in disease incidence

In Chapter 1 we found a surprising overall positive correlation between cancer and AID incidence rate sex biases across many different human tissues. Among key factors that have been previously associated with sex bias in either AID or cancer incidence, we found that the sex bias in the expression of mitochondrially encoded genes (and possibly in the expression of a few immune pathways) stands out as a key factor whose aggregate level across human tissues is quite strongly associated with these incidence rate sex biases. Our findings thus call for further mechanistic studies on the role of mitochondrial gene expression in determining cancer and AID incidence and their incidence rate sex biases. In addition to mechanistic studies, it may be possible for computational studies to build on our work as more computational strategies are devised or more sequencing data becomes available. For example, we attempted to use the Mitocalc program to estimate mitochondrial genome abundance in healthy tissues based on whole genome sequencing data [143] but were unable to find enough healthy tissue data to compare male and female samples for our tissues of interest.

Identifying translation perturbation-derived neopeptides

In Chapter 2, we demonstrated that a large-scale search for aberrant translation event products in immunoproteomics data is feasible. This allows us to perform transcriptome-wide unbiased searches such that we can study the downstream effects of translational perturbations without pre-determining restricted subsets of events for our database. False discovery rate control is possible for larger and less target-rich databases, but could be greatly improved.

We have identified several additional straightforward features we might add to improve our ability to confidently identify novel events in proteomics data. First, an explicit fragmentation coverage metric for individual PSMs could allow us to better filter out low-quality PSMs. Second, in steps where we use outside tools such as *de novo* MS sequencers of MHC binding predictors we could compare results from multiple programs. This might allow different tools to complement each other or at least highlight when one tool clearly outperforms another. Third, we could apply the pipeline (not including MHC binding prediction) to whole proteomics data. This would be advantageous because whole proteomics outputs have many more peptide spectra than immunopeptidomics outputs and thus we might have more success in detecting rare events. The whole proteomics files for the Bartok et al. study from which we retrieved our immunopeptidomics files have about twenty times more spectra for each sample than the immunopeptidomics files. We ran most of the current pipeline for these whole proteomics files but ran into scaling issues such that we will need to modify several of the modules to accommodate the increased data magnitude. Future work could include necessary code rewrites to allow us to analyze both HLA peptidomics and whole proteomics data from the same studies. Finally, we could modify our simplistic *in silico* translation strategy to take into account more

transcript features such as translation initiation sites (whether canonical or less common motifs), potential stop-codon readthrough, or transcript type (e.g. protein-coding versus unprocessed). This last step could benefit greatly from improved transcriptome annotations and more comprehensive RNA-seq datasets that would allow for more targeted and more accurate *in silico* translation.

Below-gene-level survival models

In Chapter 3 we found for the TCGA metastatic melanoma dataset that adding below-gene-level RNA-seq metrics such as isoR and isoT to multivariate Cox models greatly increases the number detected associations between tumor gene expression and patient survival. That below-gene-level models outperformed status quo gene-level models despite being more numerous (and thus facing a greater multiple hypothesis test correction penalty) suggests below-gene-level metrics are a rich source of information for survival analyses. In addition to the specific predictions for metastatic melanoma produced in our analyses we also demonstrated that the novel isoform ratio (isoR) and event TPM (eventT) outperform their close counterparts (isoform TPM and gene TPM, respectively) and so may be useful in future mRNA splicing analyses. We suspect that repeating our Chapter 3 analyses using TCGA datasets for different cancer types will be similarly informative.

mRNA splicing diversity

In Chapter 4 we introduced applications of diversity metrics from population ecology and population genetics to mRNA splicing data and showed that heteroformity is more robust to RNA-seq data depth than entropy when analyzing high splicing diversity genes. Future

applications for this work include many kinds of splicing diversity analysis, from comparing splicing changes in single genes across conditions to comparing transcriptome-wide splicing diversity changes resulting from perturbation of cellular mRNA splicing machinery.

Works Cited

[1] S. Goodwin, J. D. McPherson and W. R. McCombie, "Coming of age: ten years of next-generation sequencing technologies," *Nature Reviews Genetics,* vol. 17, pp. 333-351, 2016.

[2] J. Lonsdale, J. Thomas, M. Salvatore, R. Phillips, E. Lo, S. Shad, R. Hasz, G. Walters, F. Garcia, N. Young, B. Foster, M. Moser, E. Karasik, B. Gillard, K. Ramsey and S. Sullivan, "The Genotype-Tissue Expression (GTEx) project," *Nature Genetics,* vol. 45, pp. 580-585, 2013.

[3] C. Hutter and J. C. Zenklusen, "The Cancer Genome Atlas: Creating Lasting Value beyond Its Data," *Cell,* vol. 173, pp. 283-285, 2018.

[4] M. J. Goldman, B. Craft, M. Hastie, R. Kristupas, F. McDade, A. Kamath, A. Banerjee, Y. Luo, D. Rogers, A. N. Brooks, J. Zhu and D. Haussler, "Visualizing and interpreting cancer genomics data via the Xena platform," *Nature Biotechnology,* vol. 38, pp. 675-678, 2020.

[5] F. E. Baralle and J. Giudice, "Alternative splicing as a regulator of development and tissue identity," *Nature Reviews Molecular Cell Biology,* vol. 18, pp. 437-451, 2017.

[6] L. E. Marasco and A. R. Kornblihtt, "The physiology of alternative splicing," *Nature Reviews Molecular Cell Biology,* vol. 24, pp. 242-254, 2023.

[7] M. J. Lambert, W. O. Cochran, B. M. Wilde, K. G. Olsen and C. D. Cooper, "Evidence for widespread subfunctionalization of splice forms in vertebrate genomes," *Genome Research,* vol. 25, pp. 624-632, 2015.

[8] J. Tapial, K. C. H. Ha, T. Sterne-Weiler, A. Gohr, U. Braunschweig, A. Hermoso-Pulido, M. Quesnel-Vallières, J. Permanyer, R. Sodaei, Y. Marquez, L. Cozzuto, X. Wang, M. Gómez-Velázquez and T. Rayon, "An atlas of alternative splicing profiles and functional associations reveals new regulatory programs and genes that simultaneously express multiple major isoforms," *Genome Research,* vol. 27, pp. 1759-1768, 2017.

[9] E. T. Wang, R. Sandberg, S. Luo, I. Khrebtukova, L. Zhang, C. Mayr, S. F. Kingsmore, G. P. Schroth and C. B. Burge, "Alternative isoform regulation in human tissue transcriptomes," *Nature,* vol. 456, pp. 470-476, 2008.

[10] E. Melamud and J. Moult, "Stochastic noise in splicing machinery," *Nucleid Acids Research,* vol. 37, pp. 4873-4886, 2009.

[11] J. K. Pickrell, A. A. Pai, Y. Gilad and J. K. Pritchard, "Noisy Splicing Drives mRNA Isoform Diversity in Human Cells," *PLOS Genetics,* vol. 6, p. e1001236, 2010.

[12] M. L. Tress, F. Abascal and A. Valencia, "Alternative Splicing May Not Be the Key to Proteome Complexity," *Trends in Biochemical Sciences,* vol. 42, pp. 98-110, 2016.

[13] S. Oltean and D. O. Bates, "Hallmarks of alternative splicing in cancer," *Oncogene,* vol. 33, pp. 5311-5318, 2013.

[14] A. Sveen, S. Kilpinen, A. Ruusulehto, R. A. Lothe and R. I. Skotheim, "Aberrant RNA splicing in cancer; expression changes and driver mutations of splicing factor genes," *Oncogene,* vol. 35, pp. 2413-2427, 2016.

[15] E. El Marabti and I. Younis, "The Cancer Spliceome: Reprograming of Alternative Splicing in Cancer," *Frontiers in Molecular Biosciences,* vol. 5, p. 80, 2018.

[16] R. K. Bradley and O. Anczuków, "RNA splicing dysregulation and the hallmarks of cancer," *Nature Reviews Cancer,* vol. 23, pp. 135-155, 2023.

[17] S. C.-W. Lee and O. Abdel-Wahab, "Therapeutic targeting of splicing in cancer," *Nature Medicine,* vol. 22, pp. 976-986, 2016.

[18] L. Frankiw, D. Baltimore and G. Li, "Alternative mRNA splicing in cancer immunotherapy," *Nature Reviews Immunology,* vol. 19, pp. 675-687, 2019.

[19] Y.-S. Chang, S.-J. Tu, H.-S. Chiang, J.-C. Yen, Y.-T. Lee, H.-Y. Fang and J.-G. Chang, "Genome-Wide Analysis of Prognostic Alternative Splicing Signature and Splicing Factors in Lung Adenocarcinoma," *Genes,* vol. 11, p. 1300, 2020.

[20] S. West, S. Kumar, S. K. Batra, H. Ali and D. Ghersi, "Uncovering and characterizing splice variants associated with survival in lung cancer patients," *PLoS Computational Biology,* vol. 15, p. e1007469, 2019.

[21] Y. Zhang, L. Yan, J. Zeng, H. Zhou, H. Liu, G. Yu, W. Yao, K. Chen, Z. Ye and H. Xu, "Pan-cancer analysis opf clinical relevance of alternative splicing events in 31 human cancers," *Oncogene,* vol. 38, pp. 6678-6695, 2019.

[22] S. Shen, Y. Wang, C. Wang, Y. N. Wu and Y. Xing, "SURVIV for survival analysis of mRNA isoform variation," *Nature Communications,* vol. 7, p. 11548, 2016.

[23] A. Y. Durgeau, S. Virk, F. Corgnac and F. Mami-Chouaib, "Recent Advances in Targeting CD8 T-Cell Immunity for More Effective Cancer Immunotherapy," *Frontiers in Immunology,* vol. 9, p. 14, 2018.

[24] T. A. Chan, M. Yarchoan, E. Jaffee, C. Swanton, S. A. Quezada, A. Stenzinger and S. Peters, "Development of tumor mutation burden as an immunotherapy biomarker: utility for the oncology clinic," *Annals of Oncology,* vol. 30, pp. 44-56, 2019.

[25] R. M. Samstein, C.-H. Lee, A. N. Shoushtari, M. D. Hellmann, R. Shen, Y. Y. Janigian, 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 and Al, "Tumor mutational load predicts survival after immunotherapy across multiple cancer types," *Nature Genetics,* vol. 51, pp. 202-206, 2019.

[26] A. Sveen, S. Kilpinen, A. Ruusulehto, R. A. Lothe and R. I. Skotheim, "Aberrant RNA splicing in cancer; expression changes and driver mutations of splicing factor genes," *Oncogene,* vol. 35, pp. 2413-2427, 2015.

[27] H. Dvinge, E. Kim, O. Abdel-Wahab and R. K. Bradley, "RNA splicing factors as oncoproteins and tumor suppressors," *Nature Reviews Cancer,* vol. 16, pp. 413-430, 2016.

[28] A. C. Smart, C. A. Margolis, H. Pimentel, M. X. He, D. Miao, D. Adeegbe, T. Fugmann, K.-K. Wong and E. M. Van Allen, "Intron retention is a source of neoepitopes in cancer," *Nature Biotechnology,* vol. 36, pp. 1056-1058, 2018.

[29] J. Zhang, E. R. Mardis and C. A. Maher, "INTEGRATE-neo: a pipeline for personalized gene fusion neoantigen discovery," *Bioinformatics,* vol. 33, pp. 555-557, 2017.

[30] A. Kahles, K.-V. Lehmann, N. C. Toussaint, M. Hüser, S. G. Stark, T. Sachsenberg, O. Stegle, O. Kohlbacher and C. Sander, "Comprehensive Analysis of Alternative Splicing Across Tumors from 8,705 Patients," *Cancer Cells,* vol. 34, pp. 211-224, 2018.

[31] K. Litchfield, J. Reading, E. Lim, H. Xu, P. Liu, M. Al-Bakir, S. Wong, A. Rowan, S. Funt, T. Merghoub, M. Lauss, I. M. Svane, G. Jönsson, J. Herrero, J. Larkin, S. A. Quezada, M. D. Hellmann, S. Turajlic and C. Swanton, "Escape from nonsense mediated decay associates with anti-tumor immunogenicity," *bioRxiv,* p. https://doi.org/10.1101/823716, 14 11 2019.

[32] Y. Kong, C. M. Rose, A. A. Cass, A. G. Williams, M. Darwish, S. Lianoglou, P. M. Haverty, A.-J. Tong, C. Blanchette, M. L. Albert, I. Mellman, R. Bourgon, J. Greally, S. Jhunjhunwala and H. Chen-Harris, "Transposable element expression in tumors is associated with immune infiltration and increased antigenicity," *Nature Communications,* vol. 10, p. 5228, 2019.

[33] C. Laumont, T. Daouda, J.-P. Laverdure, E. Bonneil, O. Caron-Lizotte, M.-P. Hardy, D. P. Granados, C. Durette, S. Lemieux, P. Thibault and C. Perreault, "Global proteogenomic analysis of human MHC class I-associated peptides derived from non-canonical reading frames," *Nature Communications,* vol. 7, p. 10238, 2016.

[34] C. Laumont, K. Vincent, L. Hesnard, E. Audemard, E. Bonneil, J.-P. Laverdure, P. Gendron, M. Courcelles, M.-P. Hardy, C. Côté, C. Durette, C. St-Pierre, M. Benhammadi, J. Lanoix, S. Vobecky, E. Haddad, S. Lemieux, P. Thibault and C. Perrault, "Noncoding regions are the main source of targetable tumor-specific antigens," *Science Translational Medicine,* vol. 10, p. eaau5516, 2018.

[35] S. Tyanova, T. Temu and J. Cox, "The MaxQuant computational platform for mass spectrometry-based shotgun proteomics," *Nature Protocols,* vol. 11, pp. 2301-2319, 2016.

[36] M. Yadav, S. Jhunjhunwala, Q. T. Phung, P. Lupardus, J. Tanguay, S. Bumbaca, C. Franci, T. K. Cheung, J. Fritsche, T. Weinschenk, Z. Modrusan, I. Mellman, J. R. Lill and L. Delamarre, "Predicting immunogenic tumour mutations by combining mass spectrometry and exome sequencing," *Nature,* vol. 515, pp. 572-576, 2014.

[37] A. Gloger, D. Ritz, T. Fugmann and D. Nevi, "Mass spectrometric analysis of the HLA class I peptidome of melanoma cell lines as a promising tool for identification of putative tumor-associated HLA epitopes," *Cancer Immunology, Immunotherapy,* vol. 65, pp. 1377-1393, 2016.

[38] M. Bassani-Sternberg, E. Braulein, R. Klar, T. Engleitner, P. Sinitcyn, S. Audehm, M. Straub, J. Weber, J. Slotta-Huspenina, K. Specht, M. E. Martignoni, A. Werner, R. Hein, D. H. Busch and C. Peschel, "Direct identification of clinically relevant neoepitopes presented on native human melanoma tissue by mass spectrometry," *Nature Communications,* vol. 7, p. 13404, 2016.

[39] S. Kalaora, E. Barnea, E. Merhavi-Shoham, N. Qutob, J. K. Teer, N. Shimony, J. Schachter, S. A. Rosenberg, M. J. Besser, A. Admon and Y. Samuels, "Use of HLA peptidomics and whole exome sequencing to identify human immunogenic neo-antigens," *Oncotarget,* vol. 7, pp. 5110-5117, 2016.

[40] S. T. Ngo, F. J. Steyn and P. A. McCombe, "Gender differences in autoimmune disease," *Frontiers in Neuroendocrinology,* vol. 35, pp. 347-369, 2014.

[41] L. Moroni, I. Bianchi and A. Lleo, "Geoepidemiology, gender and autoimmune disease," *Autoimmunity Reviews,* vol. 11, pp. A386-A392, 2012.

[42] A. Clocchiatti, E. Cora, Y. Zhang and G. P. Dotto, "Sexual dimorphism in cancer," *Nature Reviews Cancer,* vol. 16, pp. 330-339, 2016.

[43] A. R. Costa, M. L. de Oliveira, I. Cruz, I. Gonçalves, J. F. Cascalheira and C. R. A. Santos, "The Sex Bias of Cancer," *Trends in Endocrinology & Metabolism,* vol. 31, pp. 785-799, 2020.

[44] S. Haupt, F. Caramia, S. L. Klein, J. B. Rubin and Y. Haupt, "Sex disparities matter in cancer developmnt and therapy," *Nature Reviews Cancer,* vol. 21, pp. 393-407, 2021.

[45] S. C. Credendino, C. Neumayer and I. Cantone, "Genetics and Epigenetics of Sex Bias: Insights from Human Cancer and Autoimmunity," *Trends in Genetics,* vol. 36, pp. 650-663, 2020.

[46] G. Edgren, L. Liming, H.-O. Adami and E. T. Chang, "Enigmatic sex disparities in cancer incidence," *European Journal of Epidemiology,* vol. 27, pp. 187-196, 2012.

[47] T. G. Consortium, "The GTEx Consortium atlas of genetic regulatory effects across human tissues," *Science,* vol. 369, pp. 1318-1330, 2020.

[48] D. o. E. a. S. A. P. D. United Nations, "World Population Prospects," 2022. [Online]. Available: https://population.un.org/wpp/Download/Standard/Population/. [Accessed 23 06 2022].

[49] F. Bray, M. Colombet, L. Mery, M. Piñeros, A. Znaor, R. Zanetti and J. Ferlay, "Cance Incidence in Five Continents," 2017. [Online]. Available: http://ci5.iarc.fr. [Accessed 18 07 2021].

[50] F. C. Registry, "Finnish Cancer Registry," 2021. [Online]. Available: https://cancerregistry.fi/statistics/cancer-statistics/. [Accessed 21 07 2021].

[51] T. S. C. Register, "The Swedish Cancer Register," 2020. [Online]. Available: https://socialstyrelsen.se/en/statistics-and-data/registers/register-information/swedish-cancer-register/. [Accessed 21 07 2021].

[52] T. C. Registry, "Taiwan Cancer Registry," 2012. [Online]. Available: http://tcr.cph.ntu.edu.tw/main.php?Page=N2. [Accessed 21 07 2021].

[53] G. Korotkevich, V. Sukhov, N. Budin, B. Shpak, M. N. Artyomov and A. Sergushichev, "Fast gene set enrichment analysis," 01 02 2021. [Online]. Available: https://doi.org/10.1101/060012.

[54] S. I. Grivennikov, F. R. Greten and M. Karin, "Immunity, inflammation, and cancer," *Cell,* vol. 140, pp. 883-899, 2010.

[55] S. L. Klein and K. L. Flanagan, "Sex differences in immune responses," *Nature Reviews Immunology,* vol. 16, pp. 626-638, 2016.

[56] C. Huang, H.-X. Zhu, Y. Yao, Z.-H. Bian, Y.-J. Zheng, L. Li, H. M. Moutsopoulos, M. E. Gershwin and Z.-X. Lian, "Immune checkpoint molecules. Possible future therapeutic implications in autoimmune diseases," *Journal of Autoimmunity,* vol. 104, p. 102333, 2019.

[57] M. Wagner, M. Jasek and L. Karabon, "Immune Checkpoint Molecules -- Inherited Variations as Markers for Cancer Risk," *Frontiers in Immunology,* vol. 11, p. 606721, 2021.

[58] A. Dunford, D. M. Weinstock, V. Savova, S. E. Schumacher, J. P. Cleary, A. Yoda, T. J. Sullivan, J. M. Hess, A. A. Gimelbrant, R. Beroukhim, M. S. Lawrence, G. Getz and A. A. Lane, "Tumor-suppressor genes that escape from X-inactivation contribute to cancer sex bias," *Nature Genetics,* vol. 49, pp. 10-16, 2016.

[59] G. Di Dalmazi, J. Hirshberg, D. Lyle, J. B. Freij and P. Caturegli, "Reactive oxygen species in organ-specific autoimmunity," *Autoimmunity Highlights,* vol. 7, p. 11, 2016.

[60] S. S. Sabharwal and P. T. Schumacker, "Mitochondrial ROS in cancer: initiators, amplifiers or an Achilles' heel?," *Nature Reviews Cancer,* vol. 14, pp. 709-721, 2014.

[61] L. Chen, B. Duvvuri, J. Grigull, R. Jamnik, J. E. Wither and G. E. Wu, "Experimental evidence that mutated-self peptides derived from mitochondrial DNA somatic mutations have the potential to trigger autoimmunity," *Human Immunology,* vol. 75, pp. 873-879, 2014.

[62] L. Hu, X. Yao and Y. Shen, "Altered mitochondrial DNA copy number contributes to human cancer risk: evidence from an updated meta-analysis," *Scientific Reports,* vol. 6, p. 35859, 2016.

[63] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander and J. P. Mesirov, "Gene set

enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *PNAS,* vol. 102, pp. 15545-15550, 2005.

[64] A. Liberzon, A. Subramanian, R. Pinchback, H. Thorvaldsdóttir, P. Tamayo and J. P. Mesirov, "Molecular signatures database (MSigDB) 3.0," *Bioinformatics,* vol. 27, pp. 1739-1740, 2011.

[65] T. Tukiainen, A.-C. Villani, A. Yen, M. A. Rivas, J. L. Marshall, R. Satija, M. Aguirre, L. Gauthier, M. Fleharty, A. Kirby, B. B. Cummings, S. E. Castel, K. J. Karczewski, F. Aguet and Byrnes, "Landscape of X chromosome inactivation across human tissues," *Nature,* vol. 550, pp. 244-248, 2017.

[66] P. J. Thul, L. Åkesson, M. Wiking, D. Mahdessian, A. Geladaki, H. A. Blal, T. Alm, A. Asplund, L. Björk, L. M. Breckels, A. Bäckström, F. Danielsson, L. Fagerberg, J. Fall, L. Gatto and C. Gnann, "A subcellular map of the human proteome," *Science,* vol. 356, p. eaal3321, 2017.

[67] D. B. Pontier and J. Gribnau, "Xist regulation and function eXplored," *Human Genetics,* vol. 130, pp. 223-236, 2011.

[68] D. N. Di Florio, J. Sin, M. J. Coronado, P. S. Atwal and D. Fairweather, "Sex differences in inflammation, redox biology, mitochondria and autoimmunity," *Redox Biology,* vol. 31, p. 101482, 2020.

[69] R. Ventura-Clapier, M. Moulin, J. Piquereau, C. Lemaire, M. Mericskay, V. Veksler and A. Garnier, "Mitochondria: a central target for sex differences in pathologies," *Clinical Science,* vol. 9, pp. 803-822, 2017.

[70] I. Mohammad, I. Starskaia, T. Nagy, J. Guo, E. Yatkin, K. Väänänen, W. T. Watford and Z. Chen, "Estrogen receptor a contributes to T cell-mediated autoimmune inflammation by promoting T cell activation and proliferation," *Science Signaling,* vol. 11, p. eeap9415, 2018.

[71] L. Papa and D. Germain, "Estrogen receptor mediates a distinct mitochondrial unfolded protein response," *Journal of Cell Science,* vol. 124, pp. 1396-1402, 2011.

[72] T. C. Kenny, A. J. Craig, A. Villanueva and D. Germain, "Mitohormesis Primes Tumor Invasion and Metastasis," *Cell Reports,* vol. 27, pp. 2292-2303.E6, 2019.

[73] T.-L. Liao, Y.-C. Lee, C.-R. Tzeng, Y.-P. Wang, H.-Y. Chang, Y.-F. Lin and S.-H. Kao, "Mitochondrial translocation of estrogen receptor B affords resistance to oxidative insult-induced apoptosis and contributes to the pathogenesis of endometriosis," *Free Radical Biology and Medicine,* vol. 134, pp. 359-373, 2019.

[74] S. Vento and F. Cainelli, "Autoimmune Diseases in Low and Middle Income Countries: A Neglected Issue in Global Health," *Israel Medical Association Journal,* vol. 18, pp. 54-55, 2016.

[75] A. Lleo, "Geoepidemiology and the Impact of Sex on Autoimmune Diseases," in *Principles of Gender-Specific Medicine*, Cambridge, MA, Academic Press, 2017.

[76] Y. Shapira, N. Agmon-Levin and Y. Shoenfeld, "Defining and analyzing geoepidemiology and human autoimmunity," *Journal of Autoimmunity,* vol. 34, pp. J168-177, 2010.

[77] F. W. Miller, L. Alfredsson, K. H. Costenbader, D. L. Kamen, L. M. Nelson, J. M. Norris and A. J. De Roos, "Epidemiology of environmental exposures and human autoimmune diseases: Findings from a National Institute of Environmental Health Sciences Expert Panel Workshop," *Journal of Autoimmunity,* vol. 39, pp. 259-271, 2012.

[78] E. Tiniakou, K. H. Costenbader and M. A. Kriegel, "Sex-specific environmental influences on the development of autoimmune diseases," *Clinical Immunology,* vol. 149, pp. 182-191, 2013.

[79] S. S. Jackson, M. A. Marks, H. A. Katki, M. B. Cook, N. Hyun, N. D. Freedman, L. L. Kahle, P. E. Castle, B. I. Graubard and A. K. Chaturvedi, "Sex disparities in the incidence of 21 cancer types: Quantification of the contribution of risk factors," *Cancer,* vol. 128, pp. 3531-3540, 2022.

[80] G. Multhoff, M. Molls and J. Radons, "Chronic inflammation in cancer development," *Frontiers in Immunology,* vol. 2, p. 98, 2012.

[81] N. Michels, C. van Aart, J. Morisse, A. Mullee and I. Huybrechts, "Chronic inflammation towards cancer incidence: A systematic review and meta-analysis of epidemiological studies," *Critical Reviews in Oncology/Hematology,* vol. 157, p. 103177, 2021.

[82] M.-m. He, C.-H. Lo, K. Wang, G. Polychronidis, L. Wang, R. Zhong, M. D. Knudsen, Z. Fang and M. Song, "Immune-Mediated Diseases Associated With Cancer Risks," *JAMA Oncology,* vol. 8, pp. 209-219, 2022.

[83] RStudio Team, "RStudio: Integrated Development for R," Boston, 2018.

[84] R Core Team, "R: A Language and Environment for Statistical Computing," Vienna, 2021.

[85] A. Kochavi, D. Lovecchio, W. J. Faller and R. Agami, "Protein diversification by mRNA translation in cancer," *Molecular Cell,* vol. 83, pp. 469-480, 2023.

[86] O. Bartok, A. Pataskar, R. Nagel, M. Laos, E. Goldfarb, D. Hayoun, R. Levy, P.-R. Körner, I. Z. M. Kreuger, J. Champagne, E. A. Zaal, O. B. Bleijerveld, X. Huang, J. Kenski and J. Wargo, "Anti-tumour immunity induces aberrant peptide presentation in melanoma," *Nature,* vol. 590, pp. 332-337, 2021.

[87] A. Pataskar, J. Champagne, R. Nagel, J. Kenski, M. Laos, J. Michaux, H. S. Pak, O. B. Bleijerveld, K. Mordente, J. M. Navarro, N. Blommaert, M. M. Nielsen, D. Lovecchio and E. Stone, "Tryptophan depletion results in tryptophan-to-phenylalanine substitutants," *Nature,* vol. 603, pp. 721-727, 2022.

[88] S.-K. Seo and B. Kwon, "Immune regulation through tryptophan metabolism," *Experimental & Molecular Medicine,* vol. 55, pp. 1371-1379, 2023.

[89] R. Ketteler, "On programmed ribosomal frameshifting: the alternative proteomes," *Frontiers in Genetics,* vol. 3, p. 242, 2012.

[90] J. D. Dinman, "Mechanisms and implications of programmed translational frameshifting," *WIREs RNA,* vol. 3, pp. 661-673, 2012.

[91] J. Champagne, K. Mordente, R. Nagel and R. Agami, "Slippy-Sloppy translation: a tale of programmed and induced-ribosomal frameshifting," *Trends in Genetics,* vol. 38, pp. 1123-1133, 2022.

[92] J. Cox and M. Mann, "MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification," *Nature Biotechnology,* vol. 26, pp. 1367-1372, 2008.

[93] A. Frank and P. Pevzner, "PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling," *Analytical Chemistry,* vol. 77, pp. 964-973, 2005.

[94] T. Kockmann and C. Panse, "The rawrr R Package: Direct Access to Orbitrap Data and Beyond," *Journal of Proteome Research,* vol. 20, pp. 2028-2034, 2021.

[95] C. Panse, J. Grossmann and S. Barkow-Oesterreicher, "Package 'protViz': Visualizing and Analyzing Mass Spectrometry Related Data in Proteomics," CRAN, 2022.

[96] J. E. Elias and S. P. Gygi, "Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry," *Nature Methods,* vol. 4, no. 3, pp. 207-214, March 2007.

[97] M. Yilmaz, W. E. Fondrie, W. Bittremieux, S. Oh and W. S. Noble, "De Novo Mass Spectrometry Peptide Sequencing with a Transformer Model," *Proceedings of the 39th International Conference on Machine Learning,* 2022.

[98] M. Yilmaz, W. E. Fondrie, W. Bittremieux, R. Nelson, V. Ananth, S. Oh and W. S. Noble, "Sequence-to-sequence translation from mass spectra to peptides with a transformer model," *bioRxiv,* 2023.

[99] A. Nesvizhskii, "Proteogenomics: concepts, applications and computational strategies," *Nature Methods,* vol. 11, pp. 1114-1125, 2014.

[100] R. Agrawal and H. V. Jagadish, "Partitioning Techniques for Large-Grained Parallelism," *IEEE Transactions on Computers,* vol. 37, pp. 1627-1634, 1988.

[101] M. Andreatta and M. Nielsen, "Gapped sequence alignment using artificial neural networks: application to the MHC class I system," *Bioinformatics,* pp. 1-7, 2015.

[102] J. Kyte and R. F. Doolittle, "A Simple Method for Displaying the Hydropathic Character of a Protein," *Journal of Molecular Biology,* vol. 157, pp. 105-132, 1982.

[103] J. Köster and S. Rahmann, "Snakemake--a scalable bioinformatics workflow engine," *Bioinformatics,* vol. 28, no. 19, pp. 2520-2522, 2012.

[104] N. Hulstaert, J. Shofstahl, T. Sachsenberg, M. Walzer, H. Barsnes, L. Martens and Y. Perez-Riverol, "ThermoRawFileParser: Modular, Scalable, and Cross-Platform RAW File Conversion," *Journal of Proteome Research,* vol. 19, pp. 537-542, 2020.

[105] A. Frankish, M. Diekhans, A.-M. Ferreira, R. Johnson, I. Jungreis, J. Loveland, J. M. Mudge, C. Sisu, J. Wright, J. Armstrong, I. Barnes, A. Berry, A. Bignell, S. C. Sala and J. Chrast, "GENCODE reference annotation for the human and mouse genomes," *Nucleic Acids Research,* vol. 8, pp. D766-D773, 2019.

[106] I.-I. Commission, *The Journal of Biological Chemistry,* 1968.

[107] B. Roy, J. D. Leszyk, D. A. Mangus and A. Jacobson, "Nonsense suppression by near-cognate tRNAs employs alternative base pairing at codon positions 1 and 3," *PNAS,* vol. 112, no. 10, pp. 3038-3043, 15 March 2015.

[108] P. Beznosková, L. Bidou, O. Namy and L. S. Valášek, "Increased expression of tryptophan and tyrosine tRNAs elevates stop codon readthrough of reporter systems in human cell lines," *Nucleic Acids Research,* vol. 49, pp. 5202-5215, 2021.

[109] D. J. Lipman and W. R. Pearson, "Rapid and Sensitive Protein Similarity Searches," *Science,* no. 227, pp. 1435-1441, 1985.

[110] J. Hundal, S. Kiwala, J. McMichael, C. A. Miller, H. Xia, A. T. Wollam, C. J. Liu, S. Zhao, Y.-Y. Feng, A. P. Graubert, A. Z. Wollam, J. Neichin, M. Neveau, J. Walker and W. Gillanders, "pVACtools: A Computational Toolkit to Identify and Visualize Cancer Neoantigens," *Cancer Immunology Research,* vol. 8, no. 3, pp. 409-420, 2020.

[111] B. Li and C. N. Dewey, "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome," *BMC Bioinformatics,* vol. 12, p. 323, 2011.

[112] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson and T. R. Gingeras, "STAR: ultrafast universal RNA-seq aligner," *Bioinformatics,* vol. 29, pp. 15-21, 2013.

[113] M. Kearse and J. E. Wilusz, "Non-AUG translation: a new start for protein synthesis in eukaryotes," *Genes & Development,* vol. 31, pp. 1717-1731, 2017.

[114] The Cancer Genome Atlas Network (TCGA), "Genomic Classification of Cutaneous Melanoma," *Cell,* vol. 161, pp. 1681-1696, 2015.

[115] M. J. Goldman, B. Craft, M. Hastie, K. Repečka, F. McDae, A. Kamath, A. Banerjee, Y. Luo, D. Rogers, A. N. Brooks, J. Zhu and D. Haussler, "Visualizing and interpreting cancer genomics data via the Xena platform," *Nature Biotechnology,* vol. 38, pp. 675-678, 2020.

[116] G. P. Alamancos, A. Pagès, J. L. Trincado, N. Bellora and E. Eyras, "Leveraging transcript quantification for fast computation of alternative splicing profiles," *RNA,* vol. 21, pp. 1521-1531, 2015.

[117] E. Lesaffre, D. Rizopoulos and R. Tsonaka, "The logistic transformation for bounded outcome scores," *Biostatistics,* vol. 8, pp. 72-85, 2007.

[118] D. Aran, M. Sirota and A. J. Butte, "Systematic pan-cancer analysis of tumour purity," *Nature Communications,* vol. 6, p. 8971, 2015.

[119] M. Bradburn, T. Clark, S. Love and D. Altman, "Survival Analysis Part II: Multivariate data analysis -- an introduction to concepts and methods," *British Journal of Cancer,* vol. 89, pp. 431-436, 2003a.

[120] M. Bradburn, T. Clark, S. Love and D. Altman, "Survival Analysis Part III: Multivariate data analysis -- choosing a model and assessing its adequacy of fit," *British Journal of Cancer,* vol. 89, pp. 605-611, 2003b.

[121] T. Clark, M. Bradburn, S. Love and D. Altman, "Survival Analysis Part I: Basic concepts and first analyses," *British Journal of Cancer,* vol. 89, pp. 232-238, 2003.

[122] J. S. Lee, A. Das, L. Jerby-Arnon, R. Arafeh, N. Auslander, M. Davidson, L. McGarry, D. James, A. Amzallag, S. G. Park, K. Cheng, W. Robinson, D. Atias, C. Stossel, E. Buzhor and G. Stein, "Harnessing synthetic lethality to predict the response to cancer treatment," *Nature Communications,* vol. 9, p. 2546, 2018.

[123] B. I. T. G. D. A. Center, "Firehose 2016_01_28 run," *Broad Institute of MIT and Harvard,* p. doi:10.7908/C11G0KM9, 2016.

[124] J. Liu, T. Lichtenberg, K. A. Hoadley, L. M. Poisson, A. J. Lazar, A. D. Cherniack, A. J. Kovatich, C. C. Benz, D. A. Levine, A. V. Lee, L. Omberg, D. M. Wolf, C. D. Shriver and V. Thorsson, "An Integrated TCGA Pan-Cancer Clinical Data Resource to Drive High-Quality Survival Outcome Analytics," *Cell,* vol. 173, pp. 400-416, 2018.

[125] J. L. Trincado, J. C. Entizne, G. Hysenai, B. Singh, M. Skalic, D. J. Elliott and E. Eyras, "SUPPA2: fast, accurate, and uncertainty-aware differential splicing analysis across multiple conditions," *Genome Biology,* vol. 19, p. 40, 2018.

[126] T. M. Therneau and P. M. Grambsch, Modeling Survival Data: Extending the Cox Model, New York: Springer, 2000.

[127] T. Therneau, "A Package for Survival Analysis in R," *R package version 3.5-5,* https://CRAN.R-project.org/package=survival.

[128] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society: Series B (Methodological),* vol. 57, pp. 289-300, 1995.

[129] A. J. Smith, P. C. Lambert and M. J. Rutherford, "Understanding the impact of sex and stage differences on melanoma cancer patient survival: a SEER-based study," *British Journal of Cancer,* vol. 124, pp. 671-677, 2020.

[130] A. Joosse, S. Collette, S. Suciu, T. Nijsten, P. M. Patel, U. Keilholz, A. M. M. Eggermont, J. W. W. Coebergh and E. de Vries, "Sex Is an Independent Prognostic Indicator for Survival and Relapse/Progression-Free Survival in Metastatized Stage III to IV Melanoma: A Pooled Analysis of Five European Organisation for Research and

Treatment of Cancer Randomized Controlled Trials," *Journal of Clinical Oncology,* vol. 31, pp. 2337-2346, 2013.

[131] H. Wickham, ggplot2: Elegant Graphics for Data Analysis, New York: Springer-Verlag, 2016.

[132] C. Ahlmann-Eltze, *Package 'ggupset': Combination Matrix Axis for 'ggplot2' to Create 'UpSet' Plots,* cran.r-project.org, 2022.

[133] L. Yan, *Package 'ggvenn': Draw Venn Diagram by 'ggplot2',* cran.r-project.org, 2023.

[134] S. Verma, D. Crawford, A. Khateb, Y. Feng, E. Sergienko, G. Pathria, C.-T. Ma, S. H. Olson, D. Scott, R. Murad, E. Ruppin, M. Jackson and Z. A. Ronai, "NRF2 mediates melanoma addiction to GCDH by modulating apoptotic signalling," *Nature Cell Biology,* vol. 24, pp. 1422-1432, 2022.

[135] M. Melé, P. G. Ferreira, F. Reverter, D. S. Deluca, J. Monlong, M. Sammeth, T. R. Young, J. M. Goldmann, D. D. Pervouchine, T. J. Sullivan, R. Johnson, A. V. Segrè, S. Djebali, A. Niarchou and T. G, "The human transcriptome across tissues and individuals," *Science,* vol. 348, pp. 660-666, 2015.

[136] L. Jost, "Entropy and diversity," *OIKOS,* vol. 113, pp. 363-375, 2006.

[137] M. Nei, "Analysis of Gene Diversity in Subdivided Populations," *PNAS,* vol. 70, pp. 3321-3323, 1973.

[138] E. H. Simpson, "Measurement of Diversity," *Nature,* vol. 163, p. 688, 1949.

[139] M. O. Hill, "Diversity and Evenness: A Unifying Notation and Its Consequences," *Ecology,* vol. 54, pp. 427-432, 1973.

[140] Y. Xie, A. Sarma, A. Vogt and Et al., *knitr: A General-Purpose Package for Dynamic Report Generation in R,* 2023.

[141] P. Jiang, S. Sinha, K. Aldape, S. Hannenhalli, C. Sahinalp and E. Ruppin, "Big data in basic and translational cancer research," *Nature Reviews Cancer,* vol. 22, pp. 625-639, 2022.

[142] W. De Coster, M. H. Weissensteiner and F. J. Sedlazeck, "Towards population-scale long-read sequencing," *Nature Reviews Genetics,* vol. 22, pp. 572-587, 2021.

[143] Y. Qian, T. J. Butler, K. Opsahl-Ong, N. S. Giroux, C. Sidore, R. Nagaraja, F. Cucca, L. Ferrucci, G. R. Abecasis, D. Schlessinger and J. Ding, "fastMitoCalc: an ultra-fast program to estimate mitochondrial DNA copy number from whole-genome sequences," *Bioinformatics,* vol. 33, pp. 1399-1401, 2017.

[144] R. Blum and C. Bresnahan, Linux Command Line and Shell Scripting Bible, Third Edition, Indianapolis: Wiley, 2015.

[145] N. Ignatiadis, B. Klaus, J. B. Zaugg and W. Huber, "Data-driven hypothesis weighting increases detection power in genome-scale multiple testing," *Nature Methods,* vol. 13, pp. 577-580, 2016.

[146] N. Ignatiadis and W. Huber, *Package 'IHW': Independent Hypothesis Weighting,* https://git.bioconductor.org/packages/IHW.

[147] S. Kalaora, E. Barnea, E. Merhavi-Shoham, N. Qutob, J. K. Teer, N. Shimony, J. Schachter, S. A. Rosenberg, M. J. Besser, A. Admon and Y. Samuels, "Use of HLA peptidomics and whole exome sequencing to identify human immunogenic neo-antigens," *Oncotarget,* vol. 7, pp. 5110-5117, 2016.

[148] S. Kalaora, Y. Wolf, T. Feferman, E. Barnea, E. Greenstein, D. Reshef, I. Tirosh, A. Reuben, S. Patkar, R. Levy, J. Quinkhardt, T. Omokoko, N. Qutob, O. Golani, J. Zhang, X. Mao and Song, "Combined Analysis of Antigen Presentation and T-cell Recognition Reveals Restricted Immune Responses in Melanoma," *Cancer Discovery,* vol. 8, pp. 1366-1375, 2018.

[149] S. Kalaora, J. S. Lee, E. Barnea, R. Levy, P. Greenberg, M. Alon, G. Yagel, G. B. Eli, R. Oren, A. Peri, S. Patkar, L. Bitton, S. A. Rosenberg, M. Lotem, Y. Levin, A. Admon and E. Ruppin, "Immunoproteasome expression is associated with better prognosis and response to checkpoint therapies in melanoma," *Nature Communications,* vol. 11, p. 896, 2020.

[150] S. Kalaora, A. Nagler, D. Nejman, M. Alon, C. Barbolin, E. Barnea, S. L. C. Ketelaars, K. Cheng, K. Vervier, N. Shental, Y. Bussi, R. Rotkopf, R. Levy, G. Benedek, S. Trabish, T. Dadosh and Levi, "Identification of bacteria-derived HLA-bound peptides in melanoma," *Nature,* vol. 592, pp. 138-143, 2021.

[151] E. L. Huttlin, A. D. Hegeman, A. C. Harms and M. R. Sussman, "Prediction of Error Associated with False-Positive Rate Determination for Peptide Identification in Large-Scale Proteomics Experiments Using a Combined Reverse and Forward Peptide Sequence Database Strategy," *Journal of Proteome Research,* vol. 6, pp. 392-398, 2007.

[152] T. J. O'Donnell, A. Rubinsteyn, M. Bonsack, A. B. Riemer, U. Laserson and J. Hammerbacker, "MHCflurry: Open-Source Class I MHC Binding Affinity Prediction," *Cell Systems,* vol. 7, pp. 129-132, 2018.

[153] A. Marcu, L. Bichmann, L. Kuchenbecker, D. J. Kowalewski, L. K. Freudenmann, L. Backert, L. Mühlenbruch, A. Szolek, M. Lübke, P. Wagner, T. Engler, S. Matovina, J. Wang and M. Hauri-Hohl, "HLA Ligand Atlas: a benign reference of HLA-presented peptides to improve T-cell-based cancer immunotherapy," *Journal for ImmunoTherapy of Cancer,* vol. 9, p. e002071, 2021.

[154] L. Gatto and K. Lilley, "MSnbase - an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation," *Bioinformatics,* vol. 28, pp. 288-289, 2012.

[155] L. Gatto, S. Gibb and J. Rainer, "MSnbase, efficient and elegant R-based processing and visualisation of raw mass spectrometry data," *bioRxiv,* 2020.

[156] A. M. Frank, M. M. Savitski, M. L. Nielsen, R. A. Zubarev and P. A. Pevzner, "De Novo Peptide Sequencing and Identification with Precision Mass Spectrometry," *Journal of Proteome Research,* vol. 6, pp. 114-123, 2007.

[157] A. M. Frank, "Predicting Intensity Ranks of Peptide Fragment Ions," *Journal of Proteome Research,* vol. 8, pp. 2226-2240, 2009.

[158] A. M. Frank, "A Ranking-Based Scoring Function for Peptide-Spectrum Matches," *Journal of Proteome Research,* vol. 8, pp. 2241-2252, 2009.

[159] N. H. Tran, X. Zhang, L. Xin, B. Shan and M. Li, "De novo peptide sequencing by deep learning," *PNAS,* vol. 114, pp. 8247-8252, 2017.

[160] B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby and G. Lajoie, "PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry," *Rapid Communications in Mass Spectrometry,* vol. 17, pp. 2337-2342, 2003.

[161] D. Kim, B. Langmead and S. L. Salzberg, "HISAT: a fast spliced aligner with lower memory requirements," *Nature Methods,* vol. 12, pp. 357-360, 2015.

[162] D. Kim, J. M. Paggi, C. Park, C. Bennett and S. L. Salzberg, "Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype," *Nature Biotechnology,* vol. 37, pp. 907-915, 2019.

[163] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin and 1000 Genome Project Data Processing Subgroup, "The Sequence Alignment/Map format and SAMtools," *Bioinformatics,* vol. 25, no. 16, pp. 2078-2079, 2009.

[164] S. Kovaka, A. V. Zimin, G. M. Pertea, R. Razaghi, S. L. Salzberg and M. Pertea, "Transcriptome assembly from long-read RNA-seq alignments with StringTie2," *Genome Biology,* vol. 20, p. 278, 2019.

[165] G. Pertea and M. Pertea, "GFF Utilities: GffRead and GffCompare," *F1000 Research,* vol. 9, p. 304, 2020.

[166] C. M. Laumont, K. Vincent, L. Hesnard, É. Audemard, É. Bonneil, J.-P. Laverdure, P. Gendron, M. Courcelles, M.-P. Hardy, C. Côté, C. Durette, C. St-Pierre, M. Benhammadi and J. Lanoix, "Noncoding regions are the main source of targetable tumor-specific antigens," *Science Translational Medicine,* vol. 10, p. eaau5516, 2018.

[167] M. Bassani-Sternberg, S. Pletscher-Frankild, L. J. Jensen and M. Mann, "Mass Spectrometry of Human Leukocyte Antigen Class I Peptidomes Reveals Strong Effects of Protein Abundance and Turnover on Antigen Presentation," *Molecular & Cellular Proteomics,* vol. 14, no. 3, pp. 658-673, 2015.

[168] T. Hirama, S. Tokita, M. Nakatsugawa, K. Murata, Y. Nannya, K. Matsuo, H. Inoko, Y. Hirohashi, S. Hashimoto, S. Ogawa, I. Takemasa, N. Sato, F. Hata, T. Kanaseki and Tor, "Proteogenomic identification of an immunogenic HLA class I neoantigen in mismatch repair-deficient colorectal cancer tissue," *JCI Insight,* vol. 6, no. 14, p. e146356, 2021.

[169] M. V. R. Cuevas, M.-P. Hardy, J. Holly, É. Bonneil, C. Durette, M. Courcelles, J. Lanoix, C. Côté, L. M. Staudt, S. Lemieux, P. Thibault, C. Perreault and J. W. Yewdell, "Most non-canonical proteins uniquely populate the proteome or immunopeptidome," *Cell Reports,* vol. 34, p. 108815, 2021.

[170] Q. Zhao, J.-P. Laverdure, J. Lanoix, C. Durette, C. Côté, É. Bonneil, C. M. Laumont, P. Gendron, K. Vincent, M. Courcelles, S. Lemieux, D. G. Millar, P. S. Ohashi and P. Thibault, "Proteogenomics Uncovers a Vast Repertoire of Shared Tumor-Specific Antigens in Ovarian Cancer," *Cancer Immunology Research,* vol. 8, pp. 544-555, 2020.

[171] J. Lehtiö, T. Arslan, I. Siavelis, Y. Pan, F. Socciarelli, O. Berkovska, H. M. Umer, G. Mermelekas, M. Pirmoradian, M. Jönsson, H. Brunnström, O. T. Brustugun, K. P. Purohit and Cunningham, "Proteogenomics of non-small cell lung cancer reveals molecular subtypes associated with specific therapeutic targets and immune-evasion mechanisms," *Nature Cancer,* vol. 2, pp. 1224-1242, 2021.

[172] C. Chong, F. Marino, H. Pak, J. Racle, R. T. Daniel, M. Müller, D. Gfeller, G. Coukos and M. Bassani-Sternberg, "High-throughput and Sensitive Immunopeptidomics Platform Reveals Profound Interferon gamma-Mediated Remodeling of the Human Leukocyte Antigen (HLA) Ligandome," *Molecular & Cellular Proteomics,* vol. 17, no. 3, pp. 533-548, 2018.

[173] C. T. Volinsky and A. E. Raftery, "Bayesian Information Criterion for Censored Survival Models," *Biometrics,* vol. 56, pp. 256-262, 2000.

[174] D. R. Crawford, S. Sinha, N. U. Nair, B. M. Ryan, J. S. Barnholtz-Sloan, S. M. Mount, A. Erez, K. Aldape, P. E. Castle, P. S. Rajagopal, C.-P. Day, A. A. Schäffer and E. Ruppin, "Sex Biases in Cancer and Autoimmune Disease Incidence Are Strongly Positively Correlated with Mitochondrial Gene Expression across Human Tissues," *Cancers,* vol. 14, p. 5885, 2022.

[175] V. Thorsson, D. L. Gibbs, S. D. Brown, D. Wolf, D. S. Bortone, T.-H. O. Yang, E. Porta-Pardo, G. F. Gao, C. L. Plaisier, J. A. Eddy, E. Ziv, A. C. Culhane, E. O. Paull, I. A. Sivakumar and Gentles, "The Immune Landscape of Cancer," *Immunity,* vol. 48, pp. 812-830, 2018.