nature portfolio

https://doi.org/10.1038/s41586-021-04244-1

Supplementary information

Valine tRNA levels and availability regulate complex I assembly in leukaemia

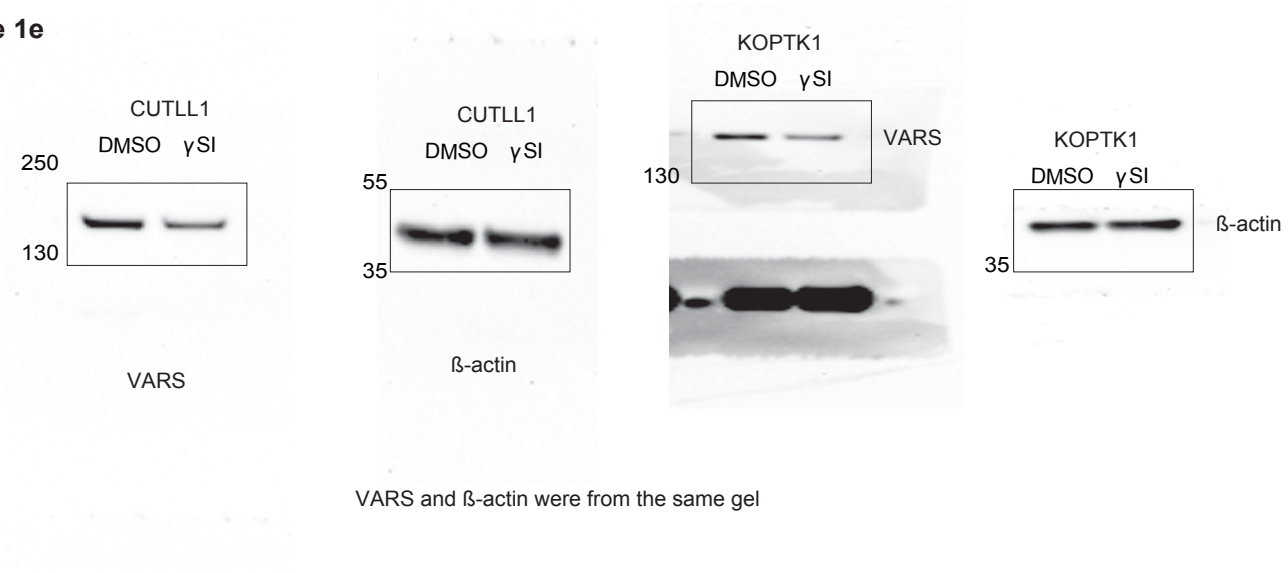
In the format provided by the authors and unedited

Valine tRNA biogenesis and bioavailability regulates complex I assembly in leukemia

Palaniraja Thandapani [1*], Andreas Kloetgen [#1,4], Matthew T. Witkowski [#1], Christina Glytsou [#1], Anna K. Lee [1], Eric Wang [1], Jingjing Wang [1], Sarah E. LeBoeuf [1], Kleopatra Avrampou [1], Thales Papagiannakopoulos[1], Aristotelis Tsirigos[1,2,3], and Iannis Aifantis[1,*]

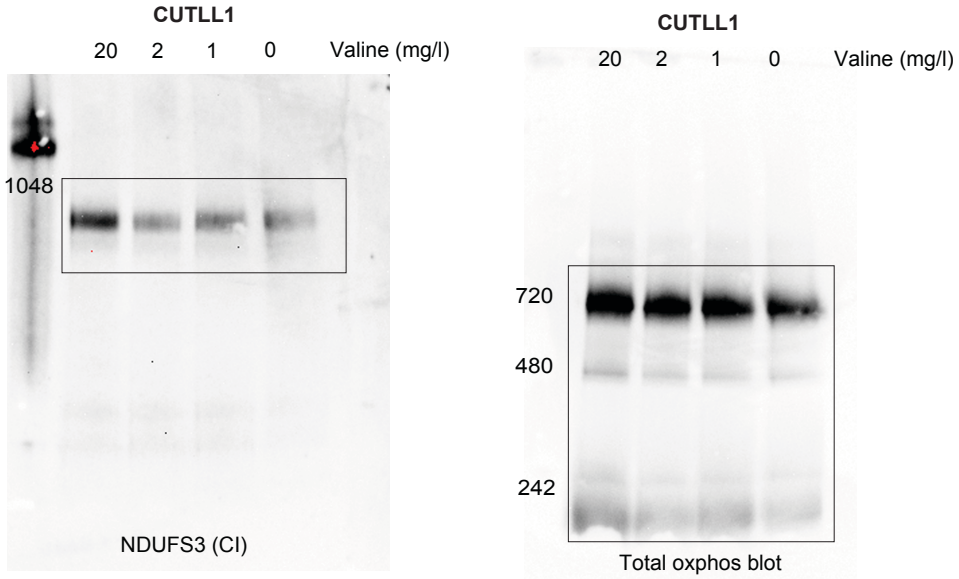
Table of contents **Page no**

Figure 1e



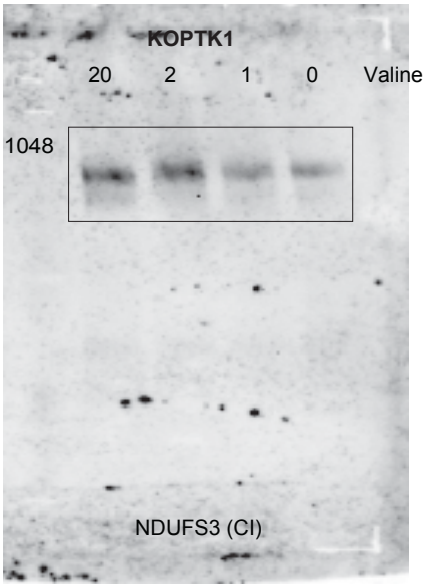
VARS and ß-actin were from the same gel

Figure 5a

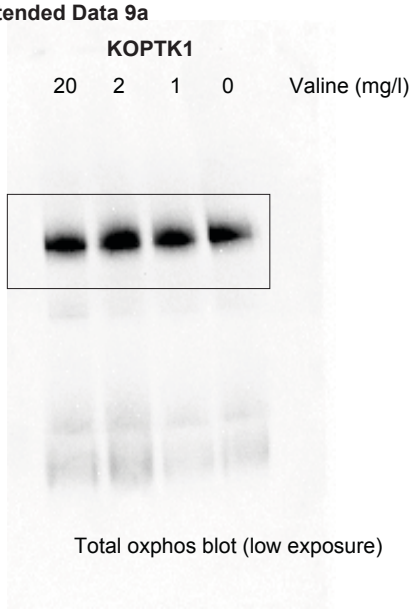


NDUFS3 and total oxphos blot were from the same gel.

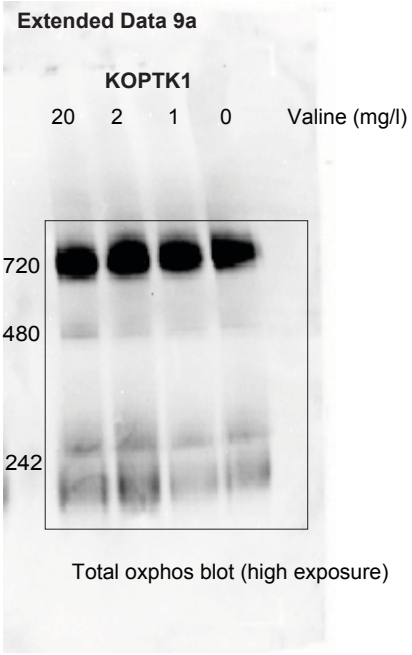
Extended Data 9a



Extended Data 9a



Extended Data 9a



NDUFS3 and total oxphos blot were from the same gel.

Figure 5c and Extended Data 9g

CUTLL1_Cas13d

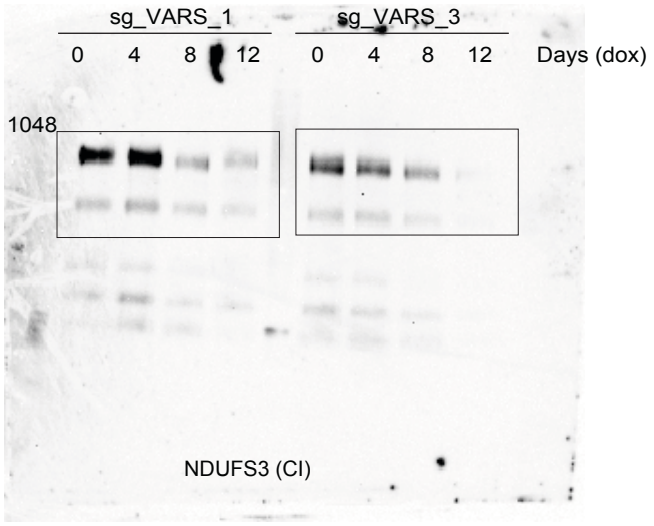


Figure 5c and Extended Data 9g

CUTLL1_Cas13d

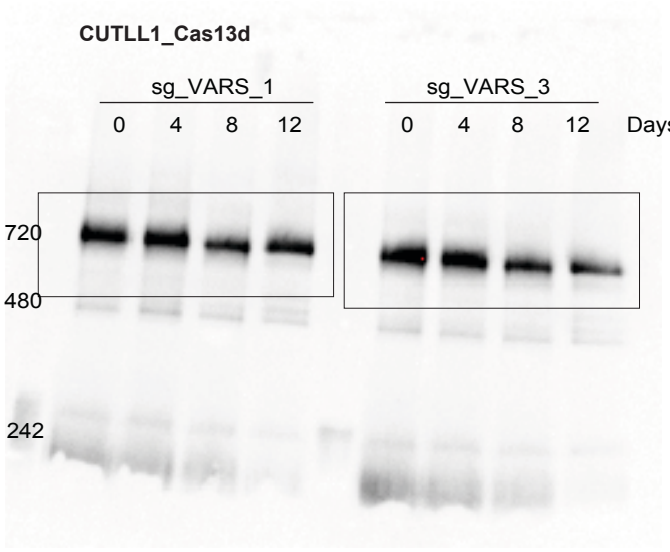
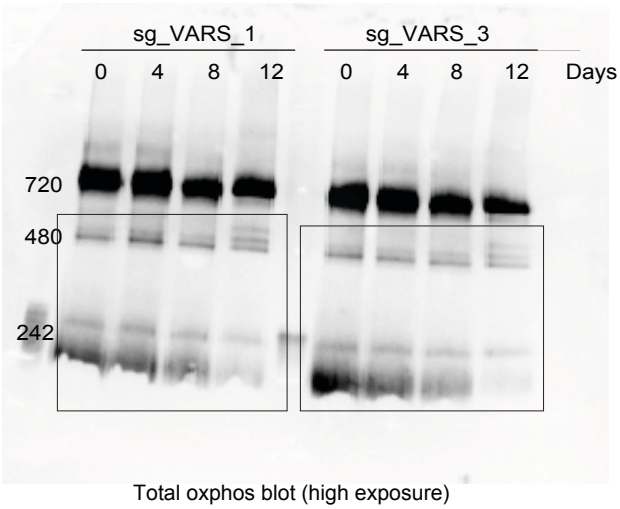


Figure 5c and Extended Data 9g

CUTLL1_Cas13d

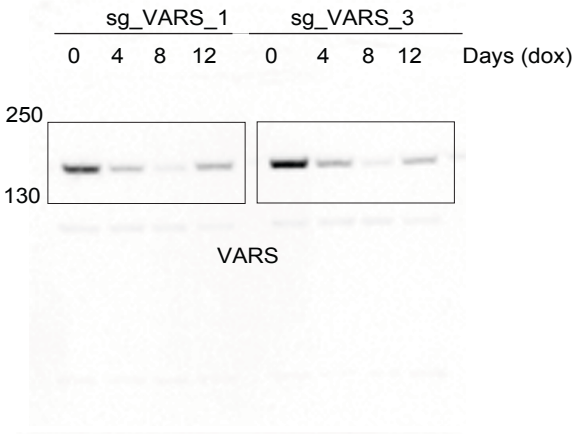


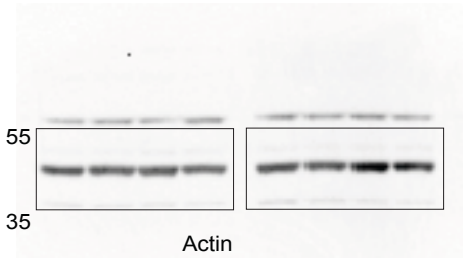
Total oxphos blot (high exposure)

NDUFS3 and total oxphos blot were from the same gel.

Figure 5c and Extended Data 9g

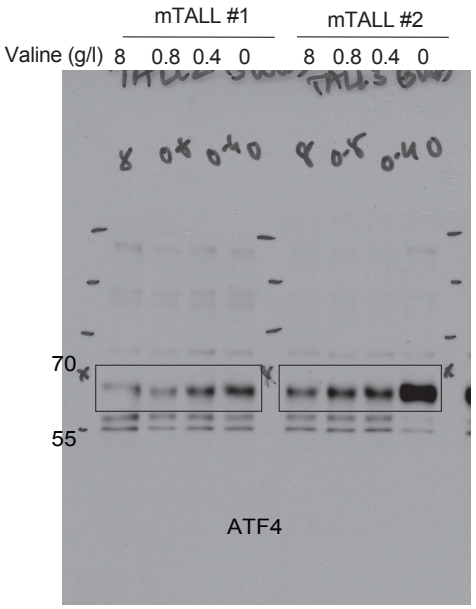
CUTLL1_Cas13d

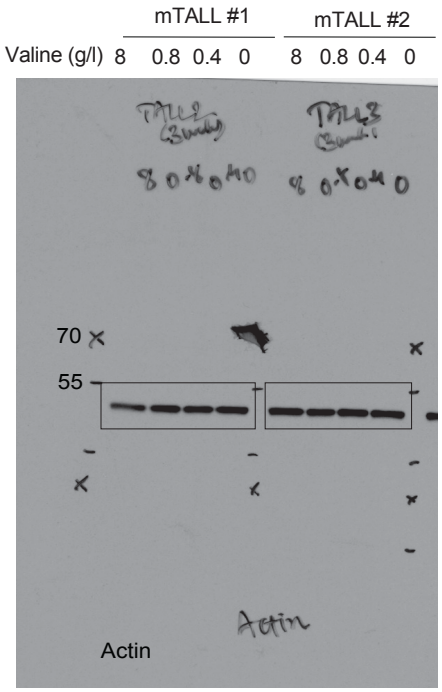




VARS and ß-actin were from the same gel

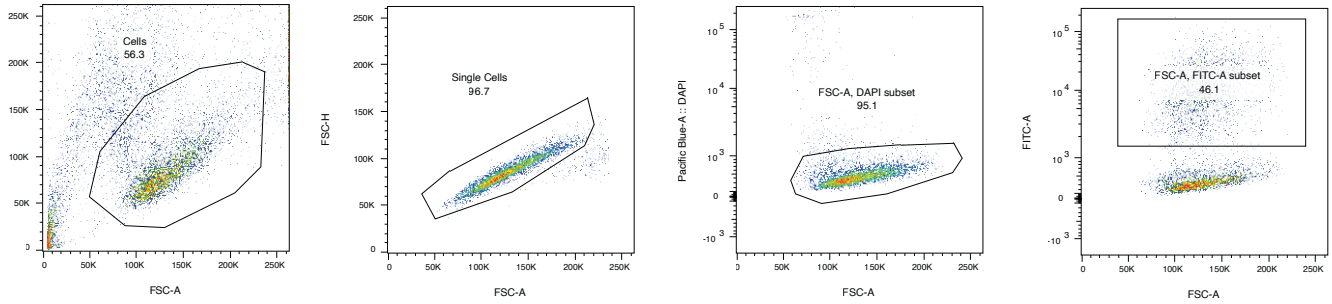
Extended Data 8l



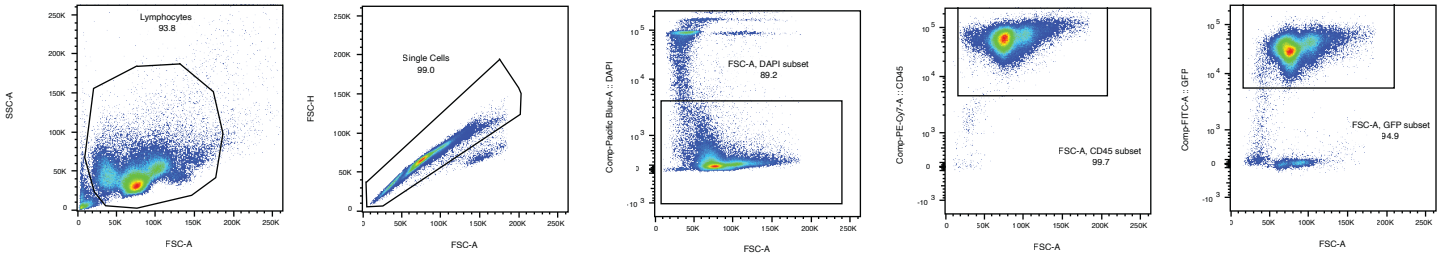


ATF4 and ß-actin were from the same gel

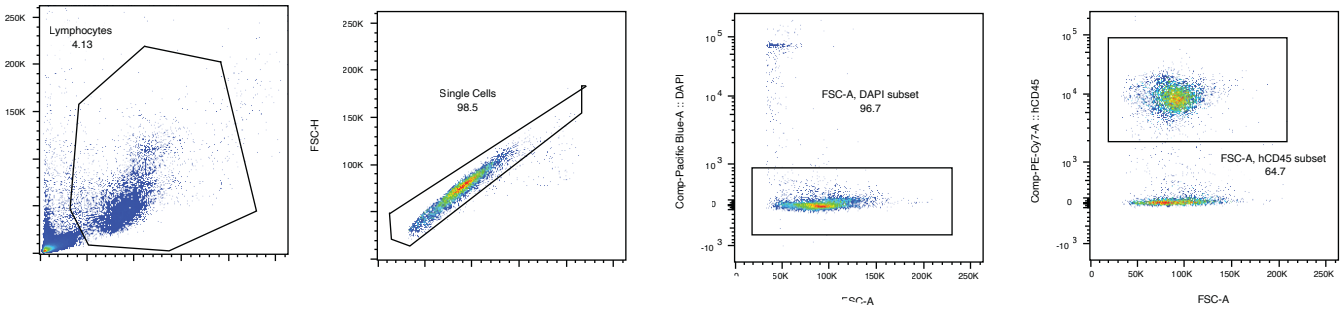
Supplementary Figure 1. This file contains the full scanned images obtained by electrophoretic separation and immunoblotting



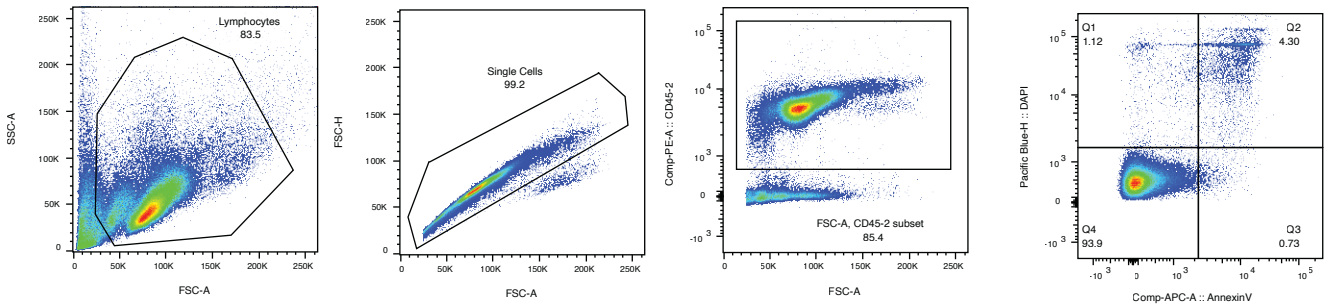
Supplemental Figure 2. Representative FACS gating strategy to determine percentage GFP+ cells over the time course of cellular fitness assays performed after knockout of candidate genes using CRISPR Cas9 or Cas13d system. Relevant to figures Extended Data 1b, 2f, 7c, 7d, 10b.



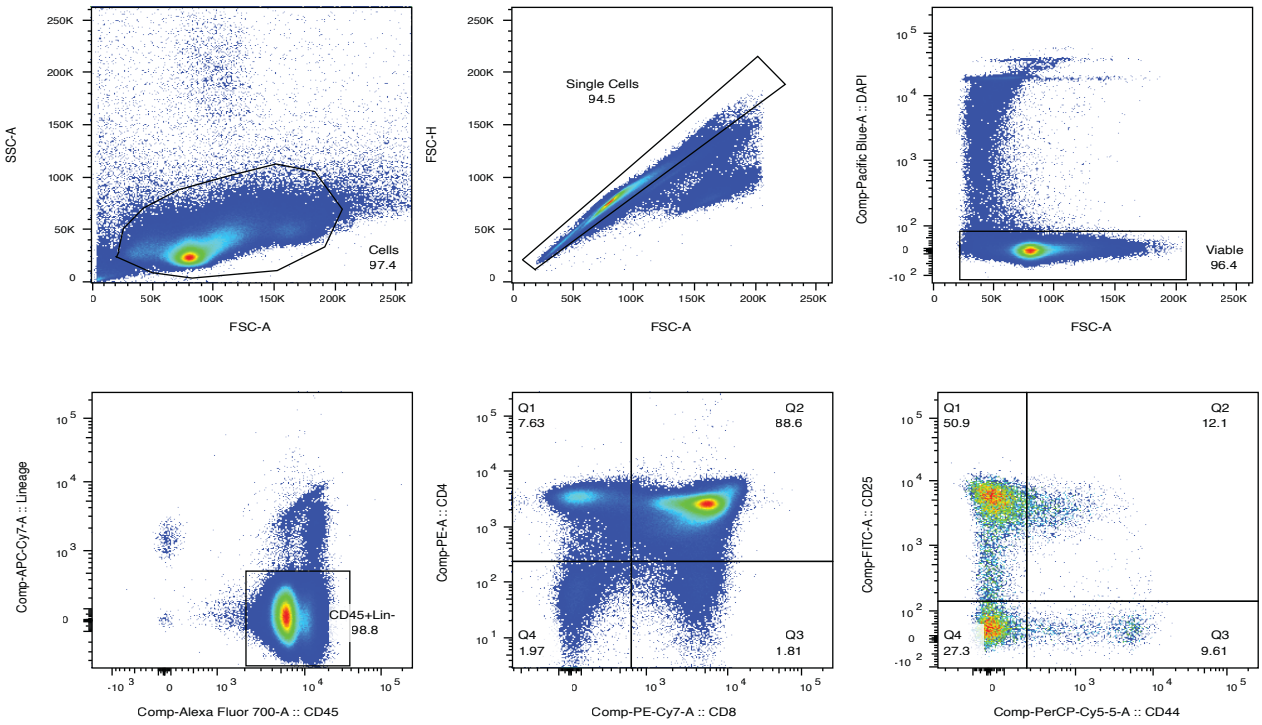
Supplemental Figure 3. Representative FACS gating strategy to determine percentage GFP+ cells *in vivo* from the peripheral bleeds of mice transplanted with primary NOTCH1-delta E-GFP tumors and fed different valine diets. Relevant to figures 2a, 2b. 2f, Extended Data 4c, 4e and 4h..



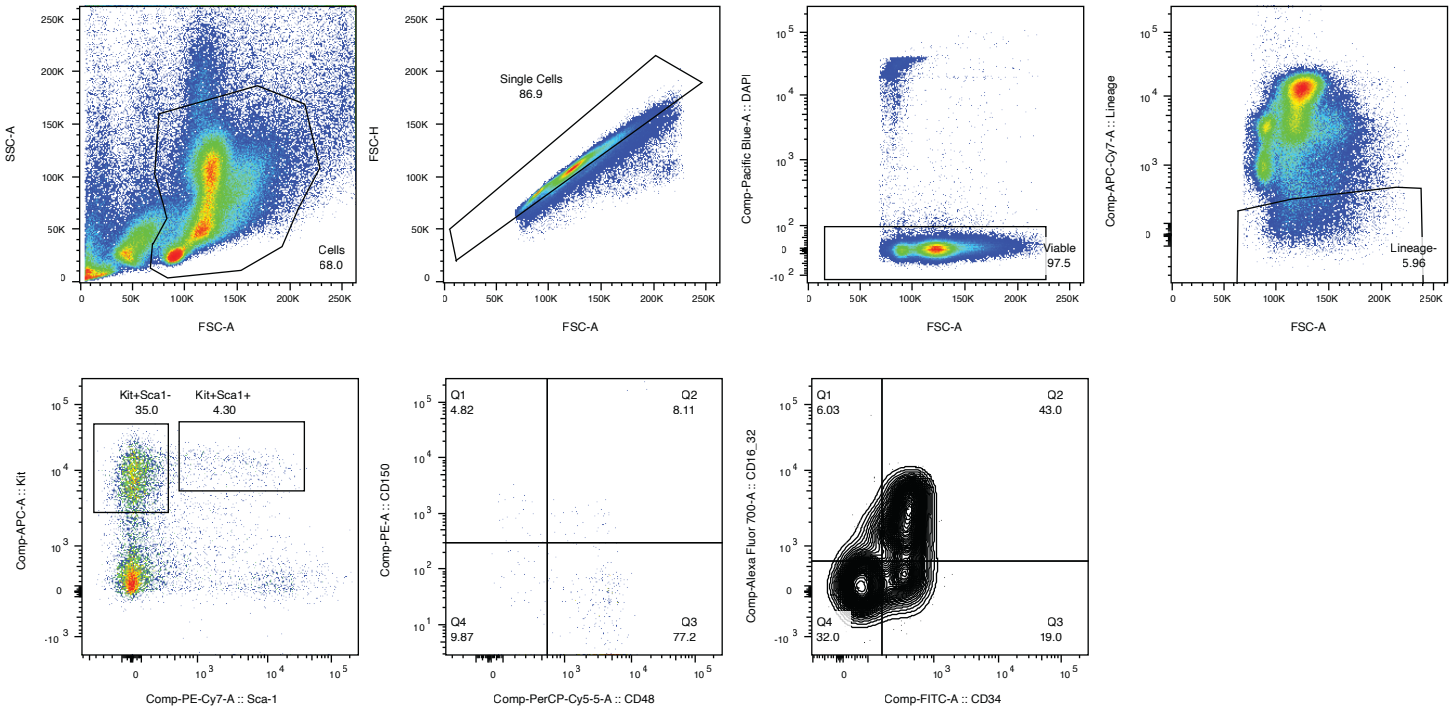
Supplemental Figure 4. Representative FACS gating strategy to determine percentage human CD45+ cells *in vivo* from the peripheral bleeds of mice transplanted with PDXs and fed different valine diets. Relevant to figures 2d.



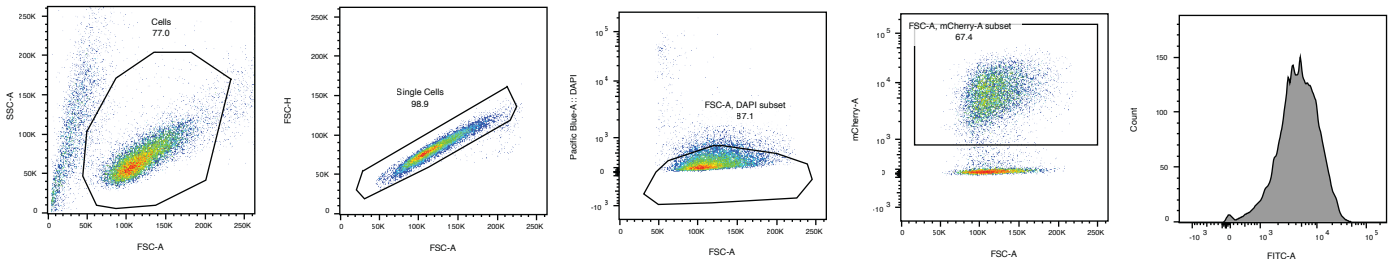
Supplemental Figure 5. Representative FACS gating strategy to determine percentage annexin V+ cells *in vivo* from the peripheral bleeds of mice transplanted with primary NOTCH1-delta E-GFP tumors and fed different valine diets. Relevant to figure Extended Data 4d. Data 1b, 2f, 7c, 7d, 10b.



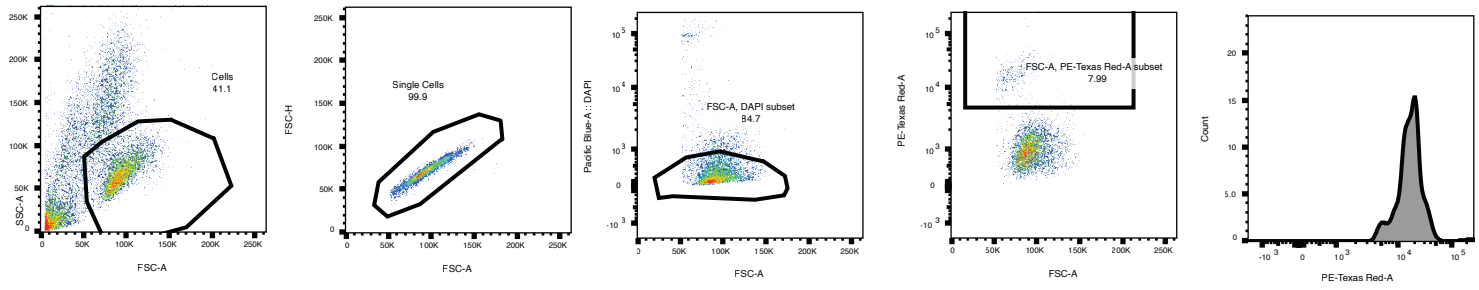
Supplemental Figure 6. Representative FACS gating strategy to determine percentage thymocyte subsets from thymus of mice fed different levels of valine. Relevant to figure Extended Data 6c..



Supplemental Figure 7. Representative FACS gating strategy to determine percentage progenitors from mice fed different levels of valine. Relevant to figure Extended Data 6a..



Supplemental Figure 8. Representative FACS gating strategy to determine GFP_d2 and mCherry mean fluorescence intensity for the reporter assays. Relevant to figure Extended Data 8j.



Supplemental Figure 9. Representative FACS gating strategy to determine mean fluorescence intensity of MitoSox Deep Red stainnig. assays. Relevant to figure Extended Data 9i.

count_gRNAs_in_CRISPR.pl

```
my $fastq = $ARGV[0];
my $guides = $ARGV[1];
my $out = $ARGV[2];

my %guides;
open(GUIDES, "<$guides");
while (my $line = <GUIDES>) {
	chomp($line);
	$line =~ s/\r//g;

	my @splitted_line = split(/[\t]/, $line);
	$splitted_line[3] = 0;
	$guides{uc($splitted_line[2])} = \@splitted_line;
}
close(GUIDES);

print "guides loaded\n";
my $rows = 0;

my $header_line = 0;
my $total_counts = 0;
my $header_line_count = 0;
open(FASTQ, "<$fastq");
while (my $line = <FASTQ>) {
	chomp($line);
	if ($header_line_count % 4 == 0) {
		$header_line = 1;
		$rows++;
	} elsif ($header_line) {
		my $sequence = substr($line, 0, 20);
		if (defined($guides{uc($sequence)})) {
			$guides{uc($sequence)}[3]++;
			$total_counts++;
		}

		$header_line = 0;
	}
	$header_line_count++;
}
close(FASTQ);

my @sorted_keys = sort {$guides{$a}[0] cmp $guides{$b}[0]} keys(%guides);

open(OUT, ">$out");
print OUT "gRNA_ID\tsymbol\tsequence\tgRNA counts\n";
foreach my $guide_seq (@sorted_keys) {
	print OUT join("\t", @{$guides{$guide_seq}}) . "\n";
}
close(OUT);
```

count_tRNA_anticodons.pl

```
my $fastq = $ARGV[0];
my $trnas = $ARGV[1];
my $out = $ARGV[2];

my %trnas;
my %trnas_counts;

open(TRNAS, "<$trnas");
while (my $line = <TRNAS>) {
	chomp($line);
	my @splitted_line = split(/\t/, $line);
	$trnas{$splitted_line[0]} = $splitted_line[1];
}
close(TRNAS);

my $header_line=0;
my $counts = 0;
my $header_line_count = 0;

open(FASTQ, "<$fastq");
while (my $line = <FASTQ>) {
	chomp($line);
	if ($header_line_count % 4 == 0) {
		$header_line = 1;
		$counts++;
	} elsif ($header_line) {
		foreach my $key (keys(%trnas)) {
			if ($line =~ /$trnas{$key}/) {
				if (defined($trnas_counts{$key})) {
					$trnas_counts{$key}++;
				} else {
					$trnas_counts{$key} = 1;
				}
			}
		}

		$header_line = 0;
	}
	$header_line_count++;
}
close(FASTQ);

open(OUT, ">$out");
foreach my $key (keys(%trnas_counts)) {
	print OUT $key . "\t" . $trnas_counts{$key} . "\n";
}
close(OUT);
```

combine_anticodons_per_aminoacid.pl

```
my $tRNAs = $ARGV[0];
my $out = $ARGV[-1];

my $ARGC = @ARGV;
my %tRNAs;

open(TRNAS, "<$tRNAs");
while (my $line = <TRNAS>) {
	chomp($line);
	my @splitted_line = split(/\t/, $line);
	my @splitted_id = split(/[-_]/, $splitted_line[0]);

	$tRNAs{$splitted_id[0] . "_" . $splitted_id[1]} = 0;
}
close(TRNAS);

for (my $i = 1; $i < $ARGC-1; $i++) {
	my $filename = $ARGV[$i];

	open(FILE, "<$filename");
	while (my $line = <FILE>) {
		chomp($line);
		my @splitted_line = split(/\t/, $line);
		my @splitted_id = split(/[-_]/, $splitted_line[0]);
		$tRNAs{$splitted_id[0] . "_" . $splitted_id[1]} += $splitted_line[1];
	}
	close(FILE);
}

my @sorted_keys = sort {$a cmp $b} keys(%tRNAs);

open(OUT, ">$out");
foreach my $key (@sorted_keys) {
	print OUT $key . "\t" . $tRNAs{$key} . "\n";
}
close(OUT);
```