Figure S1

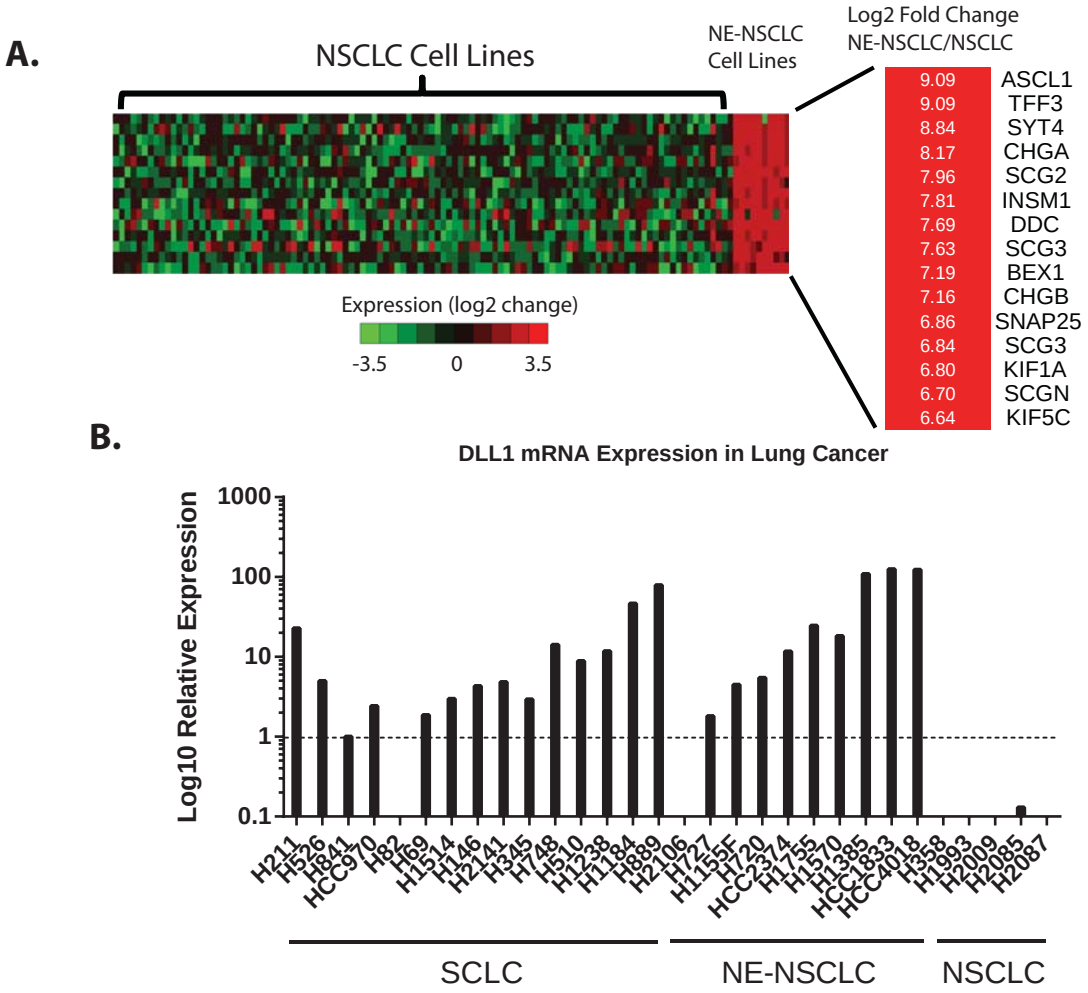


Figure S2

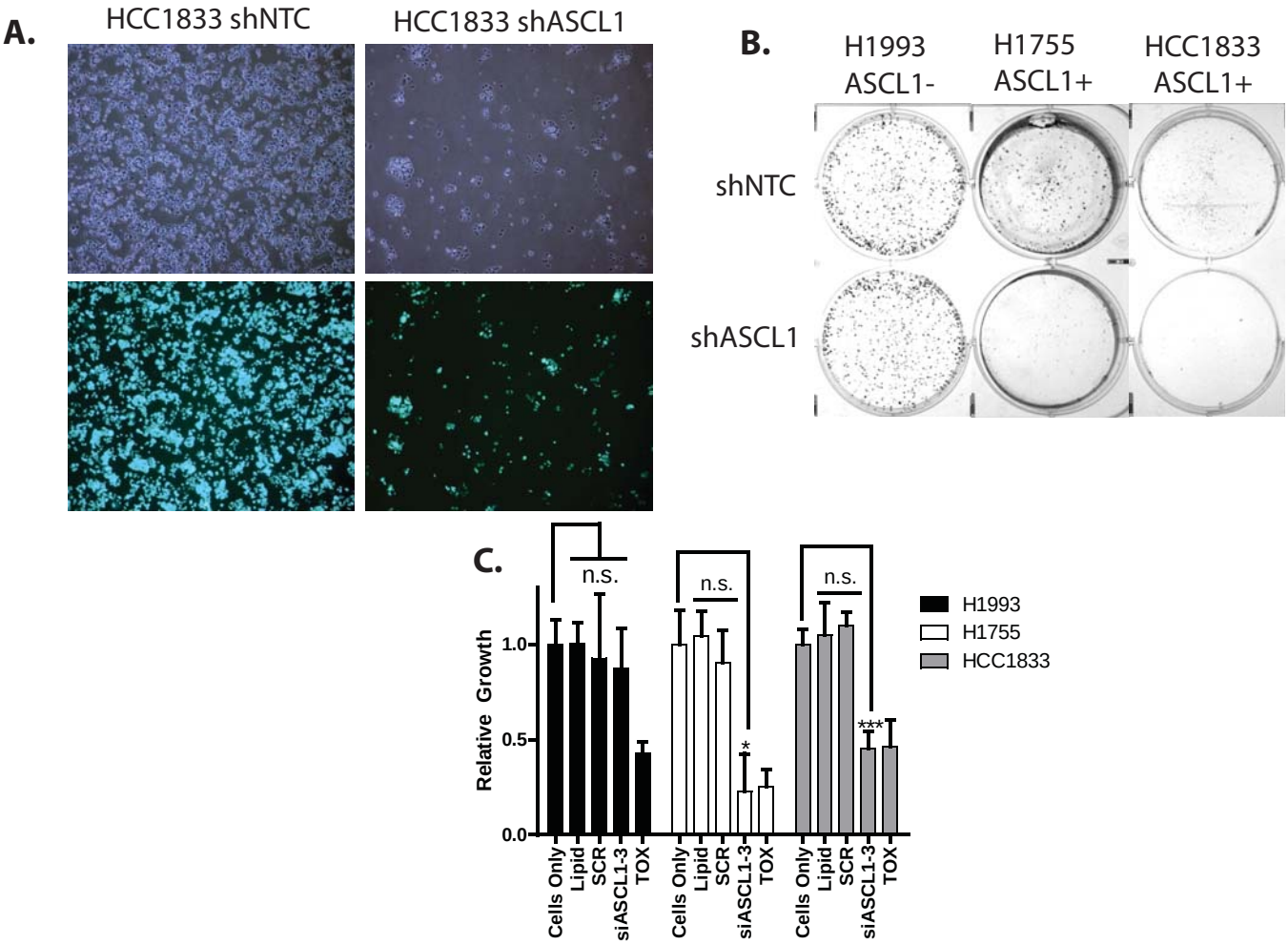


Figure S3

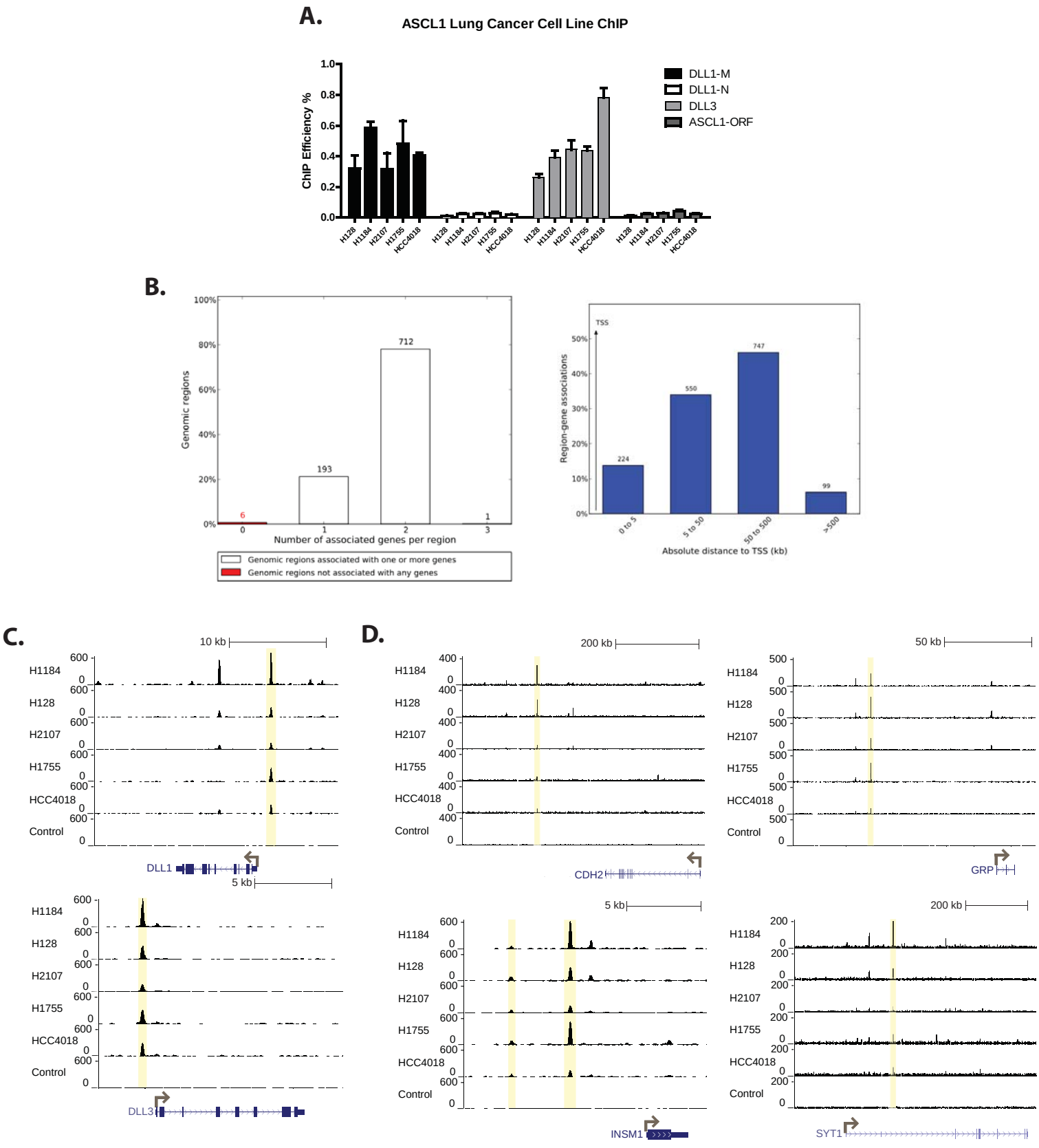

A.
ASCL1 Lung Cancer Cell Line ChIP
ChIP Efficiency %
DLL1-M
DLL1-N
DLL3
ASCL1-ORF
H128
H1184
H2107
H1755
HCC4018
B.
Genomic regions
Number of associated genes per region
Genomic regions associated with one or more genes
Genomic regions not associated with any genes
Region-gene associations
TSS
Absolute distance to TSS (kb)
0 to 5
5 to 50
50 to 500
>500
C.
D.
Control
DLL1
DLL3
CDH2
GRP
INSM1
SYT1

Figure S4


207 Lung Cell Lines
1006 ASCL1 ChIP-Seq Targets
Histology
NE-NSCLC
SCLC
NSCLC
HBEC/HSAEC
Expression
(log2 change)
6.0
0
-6.0

Figure S5

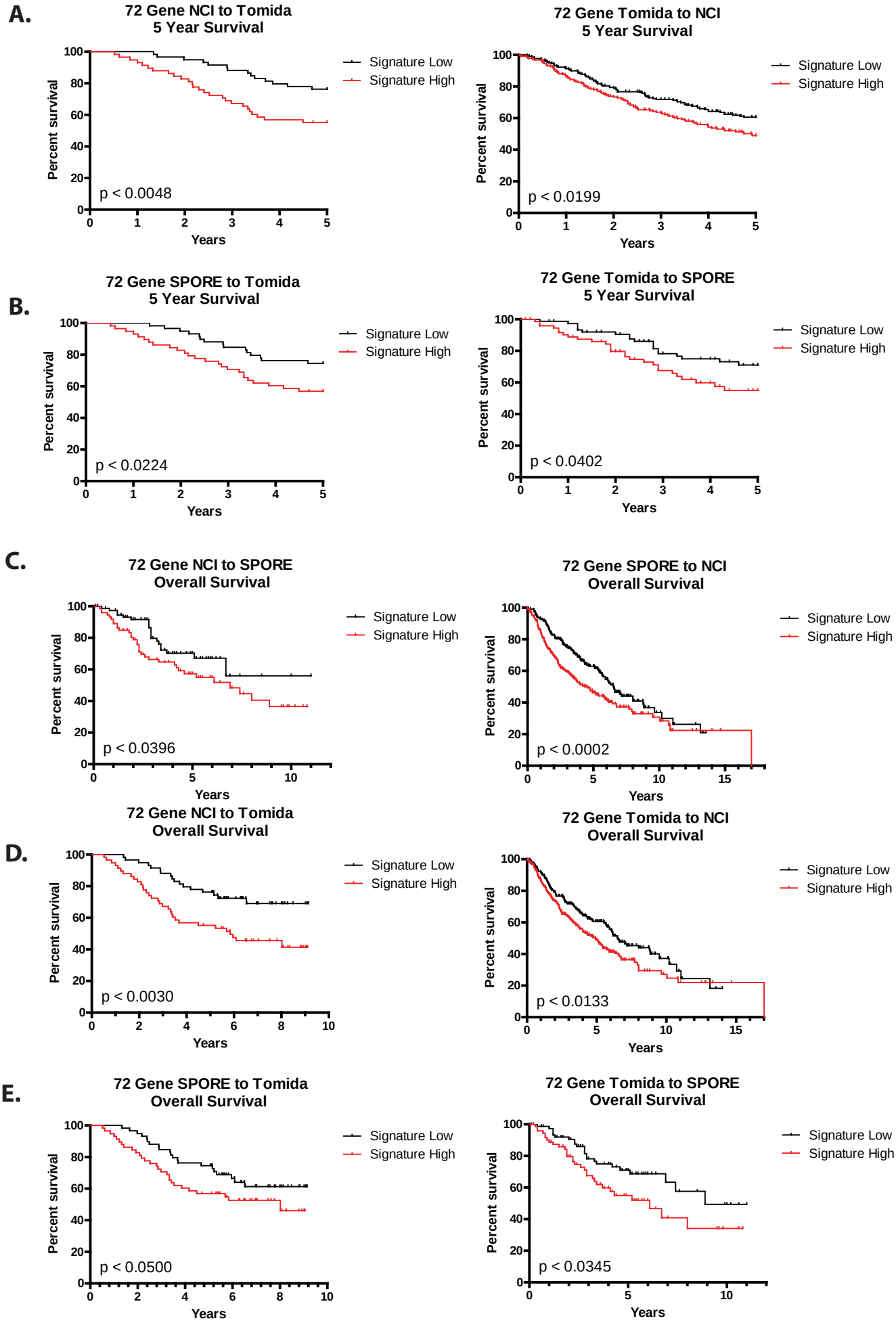
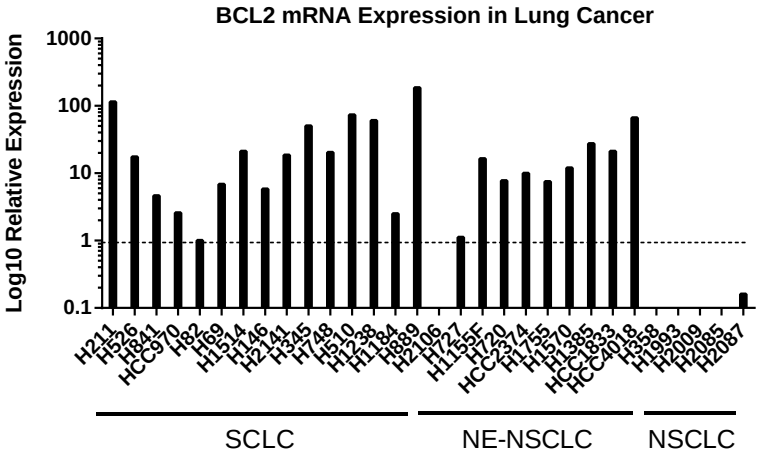

A.
72 Gene NCI to Tomida
5 Year Survival
Percent survival
Years
p < 0.0048
Signature Low
Signature High
72 Gene Tomida to NCI
5 Year Survival
p < 0.0199
B.
72 Gene SPORE to Tomida
5 Year Survival
p < 0.0224
72 Gene Tomida to SPORE
5 Year Survival
p < 0.0402
C.
72 Gene NCI to SPORE
Overall Survival
p < 0.0396
72 Gene SPORE to NCI
Overall Survival
p < 0.0002
D.
72 Gene NCI to Tomida
Overall Survival
p < 0.0030
72 Gene Tomida to NCI
Overall Survival
p < 0.0133
E.
72 Gene SPORE to Tomida
Overall Survival
p < 0.0500
72 Gene Tomida to SPORE
Overall Survival
p < 0.0345

Figure S6



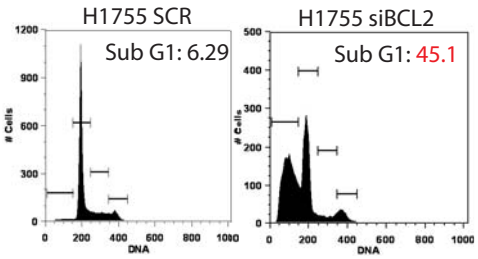
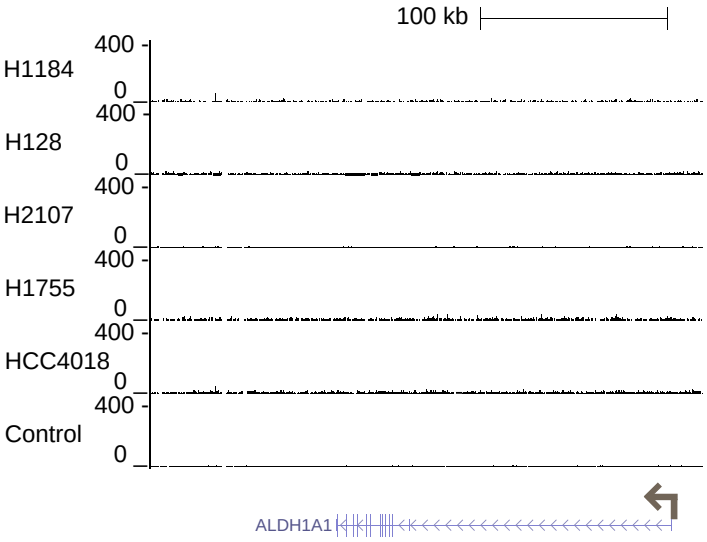
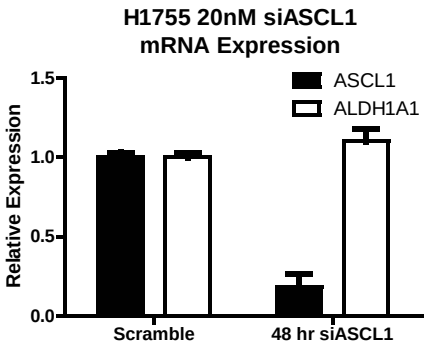


Figure S7





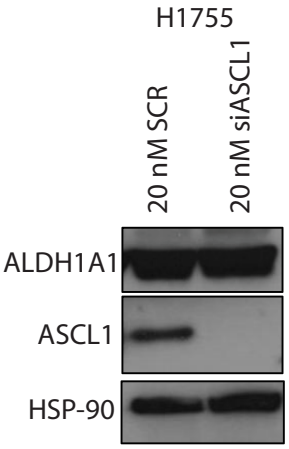


Figure S8

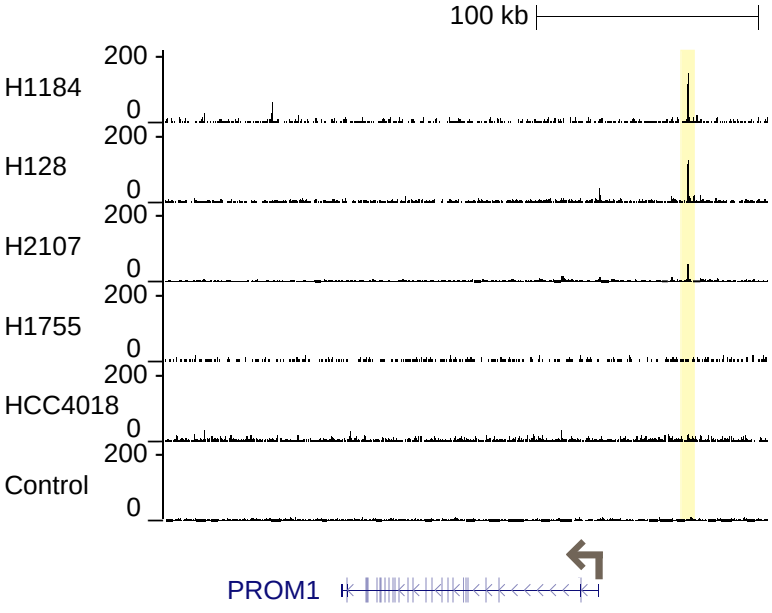
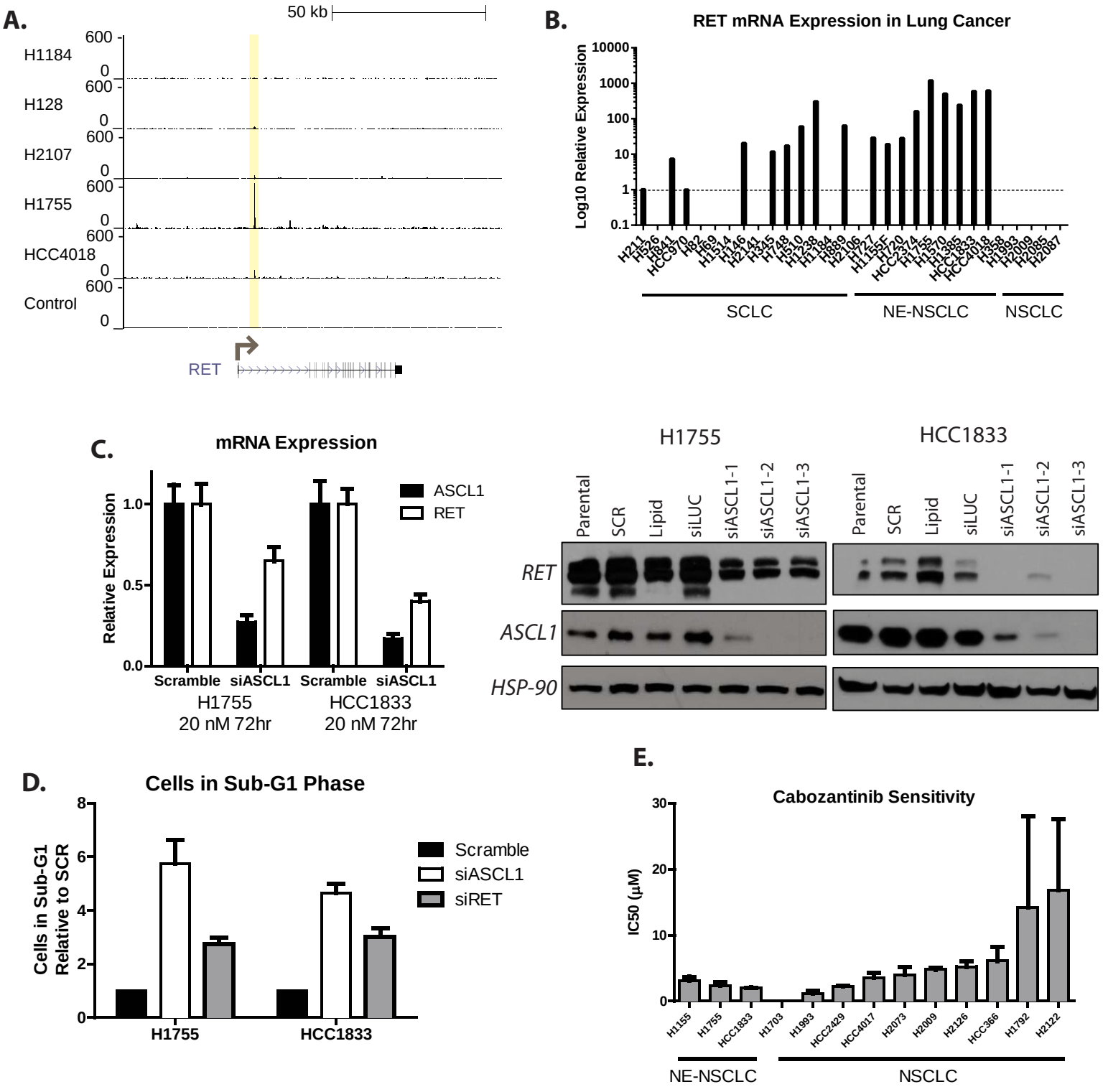


Figure S9


A.
50 kb
H1184
H128
H2107
H1755
HCC4018
Control
600
0
RET
B.
RET mRNA Expression in Lung Cancer
Log10 Relative Expression
10000
1000
100
10
1
0.1
H211
H526
H841
HCC970
H82
H69
H1514
H146
H2141
H345
H748
H510
H1238
H1184
H889
H2106
H727
H1155F
H720
HCC2374
H1755
H1570
H1385
HCC1833
HCC4018
H358
H1993
H2009
H2085
H2087
SCLC
NE-NSCLC
NSCLC
C.
mRNA Expression
Relative Expression
1.0
0.5
0.0
ASCL1
RET
Scramble
siASCL1
Scramble
siASCL1
H1755
20 nM 72hr
HCC1833
20 nM 72hr
H1755
HCC1833
Parental
SCR
Lipid
siLUC
siASCL1-1
siASCL1-2
siASCL1-3
RET
ASCL1
HSP-90
D.
Cells in Sub-G1 Phase
Cells in Sub-G1
Relative to SCR
8
6
4
2
0
Scramble
siASCL1
siRET
H1755
HCC1833
E.
Cabozantinib Sensitivity
IC50 (μM)
30
20
10
0
H1155
H1755
HCC1833
H1703
H1993
HCC2429
HCC4017
H2073
H2009
H2126
HCC366
H1792
H2122
NE-NSCLC
NSCLC

Supplementary Figure Legends

Figure S1. (A): Log2 gene expression differences between NE-NSCLC cell lines and typical NSCLC cell lines. **(B):** Delta-like 1 (DLL1) expression measured by qRT-PCR in a cohort of lung cancer cell lines. DLL1 is expressed in neuroendocrine lines including SCLC and NE-NSCLC, but not typical NSCLC.

Figure S2. (A): Stable knockdown of achaete-scute homolog 1 (ASCL1) in HCC1833 cells with shRNA reduces cell count following one week of antibiotic selection compared to non-targeting (shNTC) control. **(B):** Colony forming ability of NCI-H1755 and HCC1833-shASCL1 cells is inhibited compared to NCI-H1755 and HCC1833-shNTC cells. NCI-H1993 shows no difference in colony forming ability between shNTC and shASCL1 cells. **(C):** ASCL1 knockdown reduces growth of NE-NSCLC cell lines measured by MTS assay five days post-transfection. H1993 shows no growth difference following siASCL1-3 transfection (n = 3, t-test performed between parental and siASCL1-3-transfected cells. n.s.: not significant, * $p < 0.05$, *** $p < 0.005$).

Figure S3. (A): ChIP for ASCL1 performed on ASCL1(+) cell lines shows amplification of sequences known to bind ASCL1 such as DLL1 and DLL3. Sequences known not to bind ASCL1 do not amplify in ASCL1(+) cell lines. ChIP efficiency is graphed. **(B): Left –** Analysis using GREAT (18) shows that 713 of the consensus ASCL1 peaks are associated with more than one gene. **Right –** The majority of peak-gene interactions occur at a distance of greater than 5 kb to 1 Mb from the transcriptional start site. **(C):** ASCL1-bound peaks in the known ASCL1 target genes DLL1 and DLL3. **(D):**

Conserved ASCL1-bound peaks appear in classic neuroendocrine genes. CDH2 (NCAM), GRP, INSM1, and SYT1 are shown.

Figure S4: Supervised clustering analysis of 1330 putative ASCL1 target genes results in a grouping of neuroendocrine lung cancer cell lines (SCLC and NE-NSCLC) separately from NSCLC and HBEC/HSAEC cell lines. 1006 genes remained for clustering after filtering using Pearson Average Linkage clustering analysis.

Figure S5: 72 gene ASCL1-associated gene signature predicts for poor prognosis in multiple data sets. **(A) –** Five year survival differences in Tomida dataset trained on NCI dataset (Left, Gehan-Breslow-Wilcoxon $p < 0.0048$) and five year survival differences in NCI dataset trained on Tomida dataset (Right, Gehan-Breslow-Wilcoxon $p < 0.0199$). **(B) –** Five year survival differences in Tomida dataset trained on SPORE dataset (Left, Gehan-Breslow-Wilcoxon $p < 0.0224$) and five year survival differences in SPORE dataset trained on Tomida dataset (Right, Gehan-Breslow-Wilcoxon $p < 0.0402$). **(C) –** Overall survival differences in SPORE dataset trained on NCI dataset (Left, Gehan-Breslow-Wilcoxon $p < 0.0396$) and overall survival differences in NCI dataset trained on SPORE dataset (Right, Gehan-Breslow-Wilcoxon $p < 0.0002$). **(D) –** Overall survival differences in Tomida dataset trained on NCI dataset (Left, Gehan-Breslow-Wilcoxon $p < 0.0030$) and overall survival differences in NCI dataset trained on Tomida dataset (Right, Gehan-Breslow-Wilcoxon $p < 0.0133$). **(E) –** Overall survival differences in Tomida dataset trained on SPORE dataset (Left, Gehan-Breslow-Wilcoxon $p < 0.0500$) and overall survival differences in SPORE dataset trained on Tomida dataset (Right, Gehan-Breslow-Wilcoxon $p < 0.0345$).

Figure S6. Left – Log10 relative B-cell CLL/lymphoma 2 (BCL2) mRNA levels as measured by qRT-PCR demonstrate high expression in pulmonary neuroendocrine cancers including SCLC and NE-NSCLC, but low or absent expression in typical NSCLC cell lines. **Right –** Representative cell cycle differences in NCI-H1755 cells following knockdown of BCL2 via siRNA.

Figure S7. Aldehyde dehydrogenase 1A1 (ALDH1A1) is not a conserved transcriptional target of ASCL1. ALDH1A1 gene does not contain conserved ASCL1-bound peaks as determined by ChIP-Seq analysis. Knockdown of ASCL1 in NCI-H1755 does not result in reduced ALDH1A1 mRNA or protein expression.

Figure S8. Prominin 1 (CD133/PROM1)ChIP-Seq data demonstrates ASCL1-bound peaks only in SCLC cell lines, but not NE-NSCLC lines.

Figure S9. Ret proto-oncogene (RET) is a putative transcriptional target of ASCL1 in ASCL1(+) cell lines. **(A):** ChIP-Seq binding site in ASCL1(+) cell lines. **(B):** RET is expressed in most neuroendocrine lung cancer cell lines. **(C):** Knockdown of ASCL1 reduces *RET* mRNA and RET protein expression in NCI-H1755 and HCC1833 NE-NSCLC cell lines. **(D):** RET knockdown induces apoptosis in NE-NSCLC. **(E):** Treatment of ASCL1(+) NE-NSCLC lines and ASCL1(-) typical NSCLC lines with the RET inhibitor Cabozantinib.

Supplementary Table S1A. Gene mutations that correlate with ASCL1(+) NSCLC cell lines (number of NSCLC and NE-NSCLC lines with and without mutations listed)

Gene	NSCLC mut	NSCLC-NE mut	NSCLC wt	NE wt	p-value	p-val adj
REG3A	3	2	87	4	0.0091	0.51
PRRC2B	6	3	82	4	0.0190	0.51
SRPX	0	1	89	5	0.0190	0.51
ZMYM2	0	1	89	5	0.0190	0.51
FSHR	5	2	84	4	0.0270	0.51
CLIP1	3	2	86	5	0.0320	0.51
OR4E2	1	1	88	5	0.0340	0.51
TACR3	1	1	88	5	0.0340	0.51
TROAP	1	1	86	5	0.0440	0.51

Supplementary Table S1B. Gene mutations that correlate with ASCL1(+) NSCLC tumors from the The Cancer Genome Atlas (TCGA) (number of NSCLC and NE-NSCLC tumors with and without mutations listed)

Gene	NSCLC mut	NSCLC-NE mut	NSCLC wt	NE wt	p-value	p-val adj
REG3A	17	4	250	13	0.0280	1.0
PRRC2B	3	3	264	14	0.0032	1.0
SRPX	3	2	264	15	0.0300	1.0
ZMYM2	1	2	266	15	0.0098	1.0
FSHR	20	4	247	13	0.0440	1.0
CLIP1	6	3	261	14	0.0120	1.0
OR4E2	2	2	265	15	0.0190	1.0
TACR3	4	2	263	15	0.0440	1.0
TROAP	6	3	261	14	0.0120	1.0

Table S1. Mutation analysis of ASCL1(+) NE-NSCLC cell lines and ASCL1(+) NSCLC tumors from the TCGA. **(A):** 9 genes were identified to have significantly different mutation rates in ASCL1(+) NE-NSCLC cell lines and **(B)** ASCL1(+) TCGA tumors compared to non-ASCL1-expressing control cell lines and tumors. Significance determined by Fisher's exact test.

Supplementary Table S2. Sensitivity of NE-NSCLC cell lines to chemotherapy compared to NSCLC

Therapy	Log Ratio NE/NSCLC	T-test	Significance
Docetaxel	-0.44	0.2603	N.S.
Doxorubicin	-0.16	0.7485	N.S.
Etoposide	-0.49	0.4129	N.S.
Gemcitabine	-0.19	0.6823	N.S.
Gemcitabine/Cisplatin	0.49	0.3908	N.S.
Paclitaxel	-0.10	0.7352	N.S.
Paclitaxel/Carboplatin	-0.18	0.4968	N.S.
Pemetrexed	-1.10	0.1420	N.S.
Pemetrexed/Cisplatin	-0.42	0.4621	N.S.
Vinorelbine	-1.38	0.4037	N.S.

Table S2. NE-NSCLC cell line response to chemotherapy was compared to NSCLC. No significant differences were found between NE-NSCLC and NSCLC cell lines for standard chemotherapy regimens. Approximately 100 NSCLC cell lines were assayed for proliferation via MTS assay following addition of various chemotherapeutics, either alone or in combination. IC50 values were tabulated and utilized in this chart to separate NE-NSCLC and NSCLC sensitivities.

Supplementary Table S3. Immunohistochemical analysis of ASCL1, SRY-determining region 2 (SOX2), thyroid transcription factor 1 (TTF1) in resected adenocarcinoma and squamous cell lung cancer patient samples

IHC staining for Lineage Oncogenes			**Tumor Histology**		**Total**
ASCL1	**TTF1**	**SOX2**	**Adenocarcinoma**	**Squamous**	
+	+	+	2	0	2
+	-	+	3	1	4
+	+	-	1	0	1
+	-	-	1	0	1
-	+	+	4	0	4
-	-	+	7	27	34
-	+	-	22	0	22
-	-	-	28	5	33
				Total ASCL1(+)	8/101

Table S3. ASCL1, SOX2, and TTF1 IHC analysis of resected lung adenocarcinomas and squamous cell cancers.

Supplementary Table S4. ASCL1 ChIP-Seq data overview

Sample	Reads	Peaks	% of Peaks with Primary Motif	P-Value for Motif
Shared	--	912	87.70%	1e-509
NCI-H128	19×10^6	8,363	70.43%	1e-3768
NCI-H1184	30×10^6	10,269	81.09%	1e-4825
NCI-H2107	67×10^6	8,914	94.94%	1e-4461
NCI-H1755	34×10^6	8,395	77.08%	1e-12512
HCC4018	73×10^6	4,329	84.71%	1e-2072

Table S4. Data overview of ASCL1 ChIP-Seq experiments. “Shared” represents consensus binding peaks between NCI-H128, NCI-H1184, NCI-H2107, NCI-H1755, and HCC4018 cell lines.

Supplementary Table S5. Notch pathway representation in ASCL1 ChIP-Seq analysis

Gene	Peak Region (Distance to TSS)
DLL1	Peak 703 (-1,435)
DLL3	Peak 300 (-880)
DLL4	Peak 152 (-790)
DTX1	Peak 93 (+58,658)
DTX2	Peak 770 (-87)
HES1	Peak 623 (+196,535)
HES5	Peak 412 (-4,668)
LFNG	Peak 755 (+4,542)
NCOR2	Peak 82 (-152,878), Peak 74 (-19,415)
NOTCH1	Peak 859 (+18,932), Peak 874 (+56,923)
PSENEN	Peak 333 (-4,550)
RBPJ	Peak 658 (+131,938)

Table S5. Gene ontology analysis identifies significant enrichment of terms from the Notch pathway following ASCL1 ChIP-Seq analysis. Gene names with the location of consensus ASCL1-bound peaks are indicated.

Supplementary Table S6. Overexpressed ASCL1 ChIP-Seq target genes

72 ASCL1 Target Genes			
ASCL1	FBP1	NKAIN2	SH3BP4
BCL2	FOS	NPTX1	SLC36A4
CACNA1A	FOXA2	NR0B2	SLC6A17
CAMK1D	FOXC1	NUAK2	SMOC2
CAPS	GCA	PCNXL2	SPPL2B
CNGB1	GRP	PFKFB2	ST18
CRIP2	ID2	PLXNA2	SVIL
DGCR2	ID4	PTPRN2	TMEM61
DGKB	INA	RAB3B	TOX
DIRAS2	IRF2BP2	RGS12	TOX3
DMPK	ISG20	RNF11	TSGA10
DOCK10	KDM4B	RNF183	TTC13
DOK6	KIAA0182	RPS6KC1	WASF2
DUSP6	KRT7	SCN2A	ZBTB20
ECE1	KSR2	SCN3A	ZBTB40
ERO1LB	LYPD1	SEC11C	ZFHX3
ETS2	MAP6	SEPW1	ZNF516
FAM70B	NAV1	SETBP1	ZNF532

Table S6. 1330 target genes identified from ChIP-Seq analysis were compared to microarray expression data between ASCL1(+) NCI-H128, NCI-H1184, NCI-H2107, NCI-H1755 and HCC4018 cell lines and control ASCL1(-) NCI-H524 and NCI-H526 cell lines. 72 significantly overexpressed genes (log2 NE-NSCLC/NSCLC > 2.00, $p < 0.01$) remained and likely constitute ASCL1 transcriptional targets.

Supplementary Table S7. Potential druggable ASCL1 target genes overexpressed in neuroendocrine lung cancer cell lines.

24 Druggable and Overexpressed ASCL1 Target Genes	
BCL2	ISG20
CACNA1A	KRT7
CAMK1D	KSR2
DGKB	NR0B2
DIRAS2	NUAK2
DMPK	PFKB2
DUSP6	PTPRN2
ECE1	RPS6KC1
ERO1LB	SCN2A
ETS2	SCN3A
FOS	SLC36A4
GRP	SLC6A17

Table S7. 24 overexpressed genes from the ASCL1 ChIP-Seq analysis that are potentially druggable were culled using a database that identifies drug-gene interactions (24).

Generate the peak lists and common peak list for submission to GEO.

Tao Wang

August 13, 2013

Peak list for each sample

```
> setwd("~/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster")
> library(limma)
> h1184 = read.table("H1184.csv", sep = ",")
> h128 = read.table("H128.csv", sep = ",")
> h2107c = read.table("H2107c.csv", sep = ",")
> h4018 = read.table("H4018c.csv", sep = ",")
> h1755 = read.table("H1755.csv", sep = ",")
> labels = c("chr", "start", "end", "length", "summit", "name",
+     "p", "fold", "FDR")
> colnames(h1184) = labels
> colnames(h128) = labels
> colnames(h2107c) = labels
> colnames(h4018) = labels
> colnames(h1755) = labels
> h1184[, "name"] = "H1184"
> h128[, "name"] = "H128"
> h2107c[, "name"] = "H2107c"
> h4018[, "name"] = "H4018"
> h1755[, "name"] = "H1755"
> rownames(h1184) = paste("H1184_", rownames(h1184), sep = "")
> rownames(h128) = paste("H128_", rownames(h128), sep = "")
> rownames(h2107c) = paste("H2107_", rownames(h2107c), sep = "")
> rownames(h4018) = paste("HCC4018_", rownames(h4018), sep = "")
> rownames(h1755) = paste("H1755_", rownames(h1755), sep = "")
> h1184 = h1184[h1184[, "fold"] >= 19 & h1184[, "p"] >= 50, ]
> h128 = h128[h128[, "fold"] >= 13 & h128[, "p"] >= 50, ]
> h2107c = h2107c[h2107c[, "fold"] >= 11 & h2107c[, "p"] >= 50,
+     ]
> h4018 = h4018[h4018[, "fold"] >= 13 & h4018[, "p"] >= 50, ]
> h1755 = h1755[h1755[, "fold"] >= 12 & h1755[, "p"] >= 50, ]
> setwd("~/projects/ASCL1/data/ASCL1/submission/peaks")
```

```
> cols = c("chr", "start", "end", "summit", "p", "fold", "FDR")
> write.table(h1184[, cols], file = "peaks_H1184.txt", sep = "\t",
+     row.names = F, quote = F)
> write.table(h128[, cols], file = "peaks_H128.txt", sep = "\t",
+     row.names = F, quote = F)
> write.table(h1755[, cols], file = "peaks_H1755.txt", sep = "\t",
+     row.names = F, quote = F)
> write.table(h2107c[, cols], file = "peaks_H2107.txt", sep = "\t",
+     row.names = F, quote = F)
> write.table(h4018[, cols], file = "peaks_HCC4018.txt", sep = "\t",
+     row.names = F, quote = F)
```

Generate common peak file

```
> setwd("~/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster")
> common = read.table("peaks_A")
> common = common[, -c(2, 3)]
> colnames(common) = c("chr", "new_summit", "H1184", "H128", "H1755",
+     "H2107", "HCC4018")
> setwd("~/projects/ASCL1/data/ASCL1/submission/peaks")
> write.table(common, file = "peaks_common.txt", sep = "\t", row.names = F,
+     quote = F)
```

Draw the density plot of common peaks

Tao Wang

August 13, 2013

Write peak summits table

```
> setwd("~/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster")
> peaks = read.table("peaks_console")
> peaks = peaks[peaks[, 5] & peaks[, 6] & peaks[, 7] & peaks[,
+     8] & peaks[, 9], ]
> peaks = peaks[, c(1, 4)]
> peaks[, 2] = round(peaks[, 2]/10) * 10 + 1
> setwd("~/projects/ASCL1/data/ASCL1/figures/density")
> write.table(peaks, file = "peaks", quote = F, row.names = F,
+     col.names = F)
```

Get intensity count

```
> data = matrix(data = 0, ncol = 6, nrow = 101)
> colnames(data) = c("H1184", "H128", "H1755", "H2107", "HCC4018",
+     "Control")
> failed = 0
> for (j in 1:dim(peaks)[1]) {
+     peak = peaks[j, ]
+     for (sample in colnames(data)) {
+         cat(paste(j, sample, "\n"))
+         command = paste("grep -m 1 -P -A 50 -B 50 \"", peak[,
+             1], "\t", peak[, 2], "\t\" ~/projects/ASCL1/data/ASCL1/figures/density/",
+             sample, "_treat.txt", sep = "")
+         output = system(command, intern = T)
+         if (length(output) == 0) {
+             command = paste("grep -m 1 -P -A 70 -B 30 \"", peak[,
+                 1], "\t", peak[, 2] - 200, "\t\" ~/projects/ASCL1/data/ASCL1/figures/densi
+                 sample, "_treat.txt", sep = "")
+             output = system(command, intern = T)
+             if (length(output) == 0) {
+                 command = paste("grep -m 1 -P -A 30 -B 70 \"",
+                   peak[, 1], "\t", peak[, 2] + 200, "\t\" ~/projects/ASCL1/data/ASCL1/figu
+                   sample, "_treat.txt", sep = "")
+                 output = system(command, intern = T)
```

```
+                 if (length(output) == 0) {
+                   failed = failed + 1
+                   print(paste("failed", failed))
+                   next
+                 }
+             }
+         }
+         output = strsplit(output, split = "\t")
+         for (i in 1:101) {
+             index = (as.numeric(output[[i]][2]) - peak[, 2])/10 +
+                 51
+             if (index >= 1 && index <= 101) {
+                 data[index, sample] = data[index, sample] + as.numeric(output[[i]][3])
+             }
+         }
+     }
+ }
> print(failed)
> setwd("~/projects/ASCL1/data/ASCL1/figures/density")
> save(data, file = "density.RData")
```

Plot the average density

```
> data = as.data.frame(data)
> data$H1184 = data$H1184/0.217
> data$H128 = data$H128/0.321
> data$H1755 = data$H1755/0.191
> data$H2107 = data$H2107/1.22
> data$HCC4018 = data$HCC4018/0.308
> data = data/max(data)
> par(mar = c(6, 6, 6, 6))
> plot((-50:50) * 10, data$H1184, type = "l", col = "coral", ylim = c(0,
+     1.1), cex.lab = 1.5, lwd = 3, ylab = "Relative Peak Height",
+     xlab = "Relative Distance to Summit (bp)", cex.axis = 1.2)
> lines((-50:50) * 10, data$H128, col = "aquamarine", cex.lab = 1.5,
+     lwd = 3)
> lines((-50:50) * 10, data$H1755, col = "darkblue", cex.lab = 1.5,
+     lwd = 3)
> lines((-50:50) * 10, data$H2107, col = "darkgoldenrod1", cex.lab = 1.5,
+     lwd = 3)
> lines((-50:50) * 10, data$HCC4018, col = "darkmagenta", cex.lab = 1.5,
+     lwd = 3)
> lines((-50:50) * 10, data$Control, col = "azure4", cex.lab = 1.5,
+     lwd = 3)
> abline(v = 0, col = "green1", lwd = 4)
> lines(c(200, 260), c(0.98, 0.98), lwd = 3, col = "coral")
```

```
> text(250, 0.98, labels = "H1184", pos = 4, font = 2)
> lines(c(200, 260), c(0.91, 0.91), lwd = 3, col = "darkblue")
> text(250, 0.91, labels = "H1755", pos = 4, font = 2)
> lines(c(200, 260), c(0.84, 0.84), lwd = 3, col = "aquamarine")
> text(250, 0.84, labels = "H128", pos = 4, font = 2)
> lines(c(200, 260), c(0.77, 0.77), lwd = 3, col = "darkmagenta")
> text(250, 0.77, labels = "HCC4018", pos = 4, font = 2)
> lines(c(200, 260), c(0.7, 0.7), lwd = 3, col = "darkgoldenrod1")
> text(250, 0.7, labels = "H2107", pos = 4, font = 2)
> lines(c(200, 260), c(0.63, 0.63), lwd = 3, col = "azure4")
> text(250, 0.63, labels = "Control", pos = 4, font = 2)
```

Generating venn diagram of the overlap of ChIP-Seq peaks

Tao Wang

August 13, 2013

Set working directory and load libraries.

```
> path = "~/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster"
> library(limma)
> library(gplots)
```

Read and prepare data matrix.

```
> h1184 = read.table(paste(path, "H1184.csv", sep = "/"), sep = ",")
> h128 = read.table(paste(path, "H128.csv", sep = "/"), sep = ",")
> h2107c = read.table(paste(path, "H2107c.csv", sep = "/"), sep = ",")
> h4018 = read.table(paste(path, "H4018c.csv", sep = "/"), sep = ",")
> h1755 = read.table(paste(path, "H1755.csv", sep = "/"), sep = ",")
> labels = c("chr", "start", "end", "length", "summit", "name",
+     "p", "fold", "FDR")
> colnames(h1184) = labels
> colnames(h128) = labels
> colnames(h2107c) = labels
> colnames(h4018) = labels
> colnames(h1755) = labels
> h1184[, "name"] = paste("H1184_", rownames(h1184), sep = "")
> h128[, "name"] = paste("H128_", rownames(h128), sep = "")
> h2107c[, "name"] = paste("H2107c_", rownames(h2107c), sep = "")
> h4018[, "name"] = paste("H4018_", rownames(h4018), sep = "")
> h1755[, "name"] = paste("h1755_", rownames(h1755), sep = "")
> h1184 = h1184[h1184[, "fold"] >= 19 & h1184[, "p"] >= 50, ]
> h128 = h128[h128[, "fold"] >= 13 & h128[, "p"] >= 50, ]
> h2107c = h2107c[h2107c[, "fold"] >= 11 & h2107c[, "p"] >= 60,
+     ]
> h4018 = h4018[h4018[, "fold"] >= 13 & h4018[, "p"] >= 50, ]
> h1755 = h1755[h1755[, "fold"] >= 12 & h1755[, "p"] >= 50, ]
```

These commands prepare files that can be submitted to GREAT.

```
> write.table(file = "~/iproject/test/H1184.bed", h1184[, c("chr",
+     "start", "end", "name")], quote = F, row.names = F, col.names = F,
+     sep = "\t")
> write.table(file = "~/iproject/test/H128.bed", h128[, c("chr",
+     "start", "end", "name")], quote = F, row.names = F, col.names = F,
+     sep = "\t")
> write.table(file = "~/iproject/test/H2107.bed", h2107c[, c("chr",
+     "start", "end", "name")], quote = F, row.names = F, col.names = F,
+     sep = "\t")
> write.table(file = "~/iproject/test/H4018.bed", h4018[, c("chr",
+     "start", "end", "name")], quote = F, row.names = F, col.names = F,
+     sep = "\t")
> write.table(file = "~/iproject/test/H1755.bed", h1755[, c("chr",
+     "start", "end", "name")], quote = F, row.names = F, col.names = F,
+     sep = "\t")
```

Submit these BED files to GREAT (great.standford.edu). Read the files that are produced by GREAT

```
> h1184 = read.table("H1184.txt", skip = 1, sep = "\t")
> h128 = read.table("H128.txt", skip = 1, sep = "\t")
> h2107c = read.table("H2107c.txt", skip = 1, sep = "\t")
> h1755 = read.table("H1755.txt", skip = 1, sep = "\t")
> h4018 = read.table("H4018.txt", skip = 1, sep = "\t")
> h1184 = as.vector(unique(h1184[, 1]))
> h128 = as.vector(unique(h128[, 1]))
> h2107c = as.vector(unique(h2107c[, 1]))
> h1755 = as.vector(unique(h1755[, 1]))
> h4018 = as.vector(unique(h4018[, 1]))
```

Using the output of the GREAT webserver, we can get the 5-set venn at gene level

```
> genes = unique(c(h1184, h128, h1755, h2107c, h4018))
> venn = matrix(data = FALSE, ncol = 5, nrow = length(genes))
> rownames(venn) = genes
> colnames(venn) = c("H1184", "H128", "H1755", "H2107", "HCC4018")
> venn[h1184, "H1184"] = TRUE
> venn[h128, "H128"] = TRUE
> venn[h1755, "H1755"] = TRUE
> venn[h2107c, "H2107"] = TRUE
> venn[h4018, "HCC4018"] = TRUE
> par(mar = c(2, 2, 2, 2))
> venn = as.data.frame(venn)
> venn(venn, small = 0.6)
> write.table(venn, file = "~/iproject/test/venn_gene_level", quote = F,
+     sep = "\t")
```

And also we can get the 5 set venn at peak level

```
> setwd("/home/twang6/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster")
> venn = read.table("peaks_console")
> venn = venn[, c(5:9)]
> colnames(venn) = c("H1184", "H128", "H1755", "H2107", "H4018")
> venn = venn == 1
> venn = as.data.frame(venn)
> venn(venn)
```

Generate the bedGraph files for Genome Browser visualization

Tao Wang

August 13, 2013

Read data matrix

```
> setwd("~/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster")
> h1184 = read.table("H1184.csv", sep = ",")
> h128 = read.table("H128.csv", sep = ",")
> h2107c = read.table("H2107c.csv", sep = ",")
> h4018 = read.table("H4018c.csv", sep = ",")
> h1755 = read.table("H1755.csv", sep = ",")
> labels = c("chr", "start", "end", "length", "summit", "name",
+     "p", "fold", "FDR")
> colnames(h1184) = labels
> colnames(h128) = labels
> colnames(h2107c) = labels
> colnames(h4018) = labels
> colnames(h1755) = labels
> h1184[, "name"] = "H1184"
> h128[, "name"] = "H128"
> h2107c[, "name"] = "H2107c"
> h4018[, "name"] = "H4018c"
> h1755[, "name"] = "H1755"
> h1184 = h1184[h1184[, "fold"] >= 19 & h1184[, "p"] >= 50, ]
> h128 = h128[h128[, "fold"] >= 13 & h128[, "p"] >= 50, ]
> h2107c = h2107c[h2107c[, "fold"] >= 11 & h2107c[, "p"] >= 50,
+     ]
> h4018 = h4018[h4018[, "fold"] >= 13 & h4018[, "p"] >= 50, ]
> h1755 = h1755[h1755[, "fold"] >= 12 & h1755[, "p"] >= 50, ]
```

The normalizing constant is calculated from total tag count in each condition

```
> h1184[, "p"] = 0.217
> h128[, "p"] = 0.321
> h1755[, "p"] = 0.191
> h2107c[, "p"] = 1.22
> h4018[, "p"] = 0.308
```

```
> control = h4018[1, ]
> control[, c("name", "p")] = c("Control", 1)
```

Get the regions where the bedGraph file should be generated

```
> setwd("~/projects/ASCL1/data/ASCL1/figures/genomebrowser")
> regions = read.table("regions.txt")
> regions = as.data.frame(regions)
> colnames(regions) = c("gene", "chr", "start", "end", "pos0")
```

Construct wiggle files

```
> samples = list(control, h1184, h128, h1755, h2107c, h4018)
> for (j in 1:6) {
+     print(j)
+     sample = samples[j]
+     sample = as.data.frame(sample)
+     name = unique(sample$name)
+     scale = as.numeric(unique(sample$p))
+     temp = c()
+     for (i in 1:5) {
+         print(paste(">", i))
+         region = regions[i, ]
+         chr = as.vector(region$chr)
+         start = as.vector(region$start)
+         start = round(start/10) * 10 - 9
+         end = as.vector(region$end)
+         end = round(end/10) * 10 + 1
+         pos0 = as.vector(region$pos0)
+         pos0 = round(pos0/10) * 10 + 1
+         command = paste("grep -m 1 -P -A ", (end - pos0)/10,
+             " -B ", (pos0 - start)/10, " \"", chr, "\t", pos0,
+             "\t\" ~/projects/ASCL1/data/ASCL1/figures/density/",
+             name, "_treat.txt", sep = "")
+         output = system(command, intern = T)
+         output = strsplit(output, split = "\t")
+         for (k in 1:(1 + (end - start)/10)) {
+             line = output[[k]]
+             start_l = as.numeric(line[2])
+             num_l = as.numeric(line[3])/scale
+             if (start_l >= start && start_l <= end) {
+                 temp = rbind(temp, paste(chr, start_l, start_l +
+                   10, num_l, sep = "\t"))
+             }
+         }
+     }
+     write.table(temp, file = paste(name, ".bedGraph", sep = ""),
```

```
+         quote = F, row.names = F, col.names = F)
+ }
```

These are the headers for the bedGraph files

```
track type=bedGraph name="Control" description="Control" color=160,160,160 priority=6
track type=bedGraph name="H1184" description="H1184" color=153,50,204 priority=1
track type=bedGraph name="H128" description="H128" color=255,69,0 priority=2
track type=bedGraph name="H1755" description="H1755" color=34,139,34 priority=3
track type=bedGraph name="H2107" description="H2107" color=0,0,205 priority=4
track type=bedGraph name="HCC4018" description="HCC4018" color=102,0,51 priority=5
```

Find common peaks by hierachical clustering

Tao Wang

August 13, 2013

This R script is used to identify the common peaks in different ChIP-Seq samples by the hierachical clustering method. It only processes one chromosome at a time. A Perl script will be run to recognize this prototype R script and rewrites it to be submitted as a qsub job.

Read and prepare data matrix

```
> setwd("~/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster")
> library(limma)
> h1184 = read.table("H1184.csv", sep = ",")
> h128 = read.table("H128.csv", sep = ",")
> h2107c = read.table("H2107c.csv", sep = ",")
> h4018 = read.table("H4018c.csv", sep = ",")
> h1755 = read.table("H1755.csv", sep = ",")
> labels = c("chr", "start", "end", "length", "summit", "name",
+     "p", "fold", "FDR")
> colnames(h1184) = labels
> colnames(h128) = labels
> colnames(h2107c) = labels
> colnames(h4018) = labels
> colnames(h1755) = labels
> h1184[, "name"] = "H1184"
> h128[, "name"] = "H128"
> h2107c[, "name"] = "H2107c"
> h4018[, "name"] = "H4018"
> h1755[, "name"] = "H1755"
> rownames(h1184) = paste("H1184_", rownames(h1184), sep = "")
> rownames(h128) = paste("H128_", rownames(h128), sep = "")
> rownames(h2107c) = paste("H2107c_", rownames(h2107c), sep = "")
> rownames(h4018) = paste("H4018_", rownames(h4018), sep = "")
> rownames(h1755) = paste("h1755_", rownames(h1755), sep = "")
```

The "chr=0" line here will be recognized and replaced by a chromosome name, for example "chrX"

```
> chr = 0
```

Prepare dist matrix

```
> h1184_chr = h1184[h1184[, "chr"] == chr & h1184[, "fold"] >=
+     19 & h1184[, "p"] >= 50, ]
> h128_chr = h128[h128[, "chr"] == chr & h128[, "fold"] >= 13 &
+     h128[, "p"] >= 50, ]
> h2107c_chr = h2107c[h2107c[, "chr"] == chr & h2107c[, "fold"] >=
+     11 & h2107c[, "p"] >= 50, ]
> h4018_chr = h4018[h4018[, "chr"] == chr & h4018[, "fold"] >=
+     13 & h4018[, "p"] >= 50, ]
> h1755_chr = h1755[h1755[, "chr"] == chr & h1755[, "fold"] >=
+     12 & h1755[, "p"] >= 50, ]
> data_chr = rbind(h1184_chr, h128_chr, h2107c_chr, h4018_chr,
+     h1755_chr)
> n = dim(data_chr)[1]
> data_chr[, "summit"] = data_chr[, "summit"] + data_chr[, "start"]
> data_chr = data_chr[order(data_chr[, "summit"]), ]
> cluster = hclust(dist(data_chr[, "summit"], method = "manhattan"))
```

Cluster analysis

```
> steps = cluster$merge
> data_chr = data.frame(data_chr, matrix(data = 0, ncol = 2, nrow = n))
> colnames(data_chr)[10:11] = c("color", "con_summit")
> data_chr[, "color"] = c(-1:-n)
> for (i in 1:(n - 1)) {
+     operations = c()
+     for (j in 1:2) {
+         if (steps[i, j] < 0) {
+             operations = c(operations, -steps[i, j])
+         }
+         if (steps[i, j] > 0) {
+             if (dim(data_chr[data_chr[, "color"] == steps[i,
+                 j], ])[1] > 0) {
+                 operations = c(operations, which(data_chr[, "color"] ==
+                   steps[i, j]))
+             }
+             else {
+                 operations = -1
+             }
+         }
+     }
+     if (operations > 0 && sum(table(data_chr[operations, "name"]))/length(table(data_chr[o
+         "name"])) == 1 && max(data_chr[operations, "summit"]) -
+         min(data_chr[operations, "summit"]) < 300) {
+         data_chr[operations, "color"] = i
+     }
```

```
+ }
> colors = names(table(data_chr[, "color"]))
```

Assign summits

```
> for (color in colors) {
+     temp = data_chr[data_chr[, "color"] == color, ]
+     summit = (temp[, "summit"] %*% sqrt(temp[, "fold"]))/sum(sqrt(temp[,
+         "fold"]))
+     data_chr[data_chr[, "color"] == color, "con_summit"] = as.numeric(round(summit))
+ }
```

Get a venn count of number of peaks falling into each region

```
> con_data = as.data.frame(matrix(data = 0, nrow = length(colors),
+     ncol = 9))
> rownames(con_data) = colors
> colnames(con_data) = c("chr", "new_start", "new_end", "summit",
+     "H1184", "H128", "H1755", "H2107c", "H4018")
> con_data[, "chr"] = chr
> for (color in colors) {
+     temp = data_chr[data_chr[, "color"] == color, ]
+     con_data[color, "summit"] = unique(temp[, "con_summit"])
+     con_data[color, temp[, "name"]] = temp[, "summit"]
+ }
> con_data[, "new_start"] = con_data[, "summit"] - 39
> con_data[, "new_end"] = con_data[, "summit"] + 40
> venn = vennCounts(con_data[, c("H1184", "H128", "H1755", "H2107c",
+     "H4018")])
```

Write result into flat files

```
> setwd("~/projects/ASCL1/data/ASCL1/chipseq/macs14/final_runs/by_cluster")
> keep_H1184 = con_data[, "H1184"] > 0
> keep_H128 = con_data[, "H128"] > 0
> keep_H1755 = con_data[, "H1755"] > 0
> keep_H2107c = con_data[, "H2107c"] > 0
> keep_H4018 = con_data[, "H4018"] > 0
> write.table(con_data, file = paste(chr, "peaks", "console", sep = "_"),
+     quote = FALSE, row.names = FALSE, col.names = FALSE, sep = "\t")
> write.table(con_data[keep_H1184 & keep_H128 & keep_H1755 & keep_H2107c &
+     keep_H4018, ], file = paste(chr, "peaks", "A", sep = "_"),
+     quote = FALSE, row.names = FALSE, col.names = FALSE, sep = "\t")
> write.table(con_data[keep_H1184 & keep_H128 & (!keep_H1755) &
+     keep_H2107c & (!keep_H4018), ], file = paste(chr, "peaks",
+     "B", sep = "_"), quote = FALSE, row.names = FALSE, col.names = FALSE,
+     sep = "\t")
```

```
> write.table(con_data[(!keep_H1184) & (!keep_H128) & keep_H1755 &
+     (!keep_H2107c) & keep_H4018, ], file = paste(chr, "peaks",
+     "C", sep = "_"), quote = FALSE, row.names = FALSE, col.names = FALSE,
+     sep = "\t")
> write.table(con_data[keep_H1184 & keep_H128 & keep_H2107c, ],
+     file = paste(chr, "peaks", "AB+", sep = "_"), quote = FALSE,
+     row.names = FALSE, col.names = FALSE, sep = "\t")
> write.table(con_data[keep_H1755 & keep_H4018, ], file = paste(chr,
+     "peaks", "AC+", sep = "_"), quote = FALSE, row.names = FALSE,
+     col.names = FALSE, sep = "\t")
> write.table(con_data[keep_H1184 & (!keep_H128) & (!keep_H1755) &
+     (!keep_H2107c) & (!keep_H4018), ], file = paste(chr, "peaks",
+     "D", sep = "_"), quote = FALSE, row.names = FALSE, col.names = FALSE,
+     sep = "\t")
> write.table(con_data[(!keep_H1184) & keep_H128 & (!keep_H1755) &
+     (!keep_H2107c) & (!keep_H4018), ], file = paste(chr, "peaks",
+     "E", sep = "_"), quote = FALSE, row.names = FALSE, col.names = FALSE,
+     sep = "\t")
> write.table(con_data[(!keep_H1184) & (!keep_H128) & keep_H1755 &
+     (!keep_H2107c) & (!keep_H4018), ], file = paste(chr, "peaks",
+     "F", sep = "_"), quote = FALSE, row.names = FALSE, col.names = FALSE,
+     sep = "\t")
> write.table(con_data[(!keep_H1184) & (!keep_H128) & (!keep_H1755) &
+     keep_H2107c & (!keep_H4018), ], file = paste(chr, "peaks",
+     "G", sep = "_"), quote = FALSE, row.names = FALSE, col.names = FALSE,
+     sep = "\t")
> write.table(con_data[(!keep_H1184) & (!keep_H128) & (!keep_H1755) &
+     (!keep_H2107c) & keep_H4018, ], file = paste(chr, "peaks",
+     "H", sep = "_"), quote = FALSE, row.names = FALSE, col.names = FALSE,
+     sep = "\t")
```

Cluster patients into different groups based on expression data

Tao Wang

August 13, 2013

Read datasets. Choose one at a time

```
> dataset = "mda"
> setwd("~/projects/ASCL1/code/ASCL1/expression/survival")
> source("survival_plot.R")
> exp = read.datasets(dataset)$exp
> setwd("~/projects/ASCL1/data/ASCL1/expression/survival")
> gene_names = read.table("Alex_72.txt", stringsAsFactors = F)[,
+     1]
> eset = exp[exp$S %in% gene_names, ]
> eset = aggregate(eset[, -1], by = list(eset[, 1]), mean)
```

Cluster patients

```
> par(mar = c(3, 3, 3, 3))
> clust = hclust(dist(t(as.matrix(eset[, -1]))))
> plot(clust, cex = 0.5, main = dataset, xlab = "patients")
```

Write expression data into a table. Rows and columns should be ordered as in the heatmap

```
> rownames(eset) = eset[, 1]
> eset = as.matrix(eset[, -1])
> for (i in 1:dim(eset)[1]) {
+     eset[i, ] = eset[i, ] - mean(eset[i, ])
+ }
> setwd("~/projects/ASCL1/data/ASCL1/expression/microarray/cluster")
> pdf(file = paste(dataset, ".pdf", sep = ""))
> hm = heatmap(eset, cexCol = 0.2)
> dev.off()
```

```
RStudioGD
        2
```

```
> eset = eset[, colnames(eset)[hm$col]]
> eset = eset[rownames(eset)[hm$row], ]
> write.csv(eset, file = paste(dataset, ".csv", sep = ""), quote = F)
```

KM-plot of the survival of adenocarcinoma patients

Tao Wang

August 13, 2013

Define the function to read patient/expression datasets

```
> read.datasets = function(dataset) {
+     setwd("~/iproject/survival")
+     if (dataset == "mda") {
+         pat = read.csv("MDA209_patient.csv", as.is = T)
+         pat$stage = NA
+         pat$stage[grep("I", pat$stage.title)] = 1
+         pat$stage[grep("II|III|IV", pat$stage.title)] = 2
+         pat = pat[pat$Diag == "Adenocarcinoma", ]
+         pat = pat[, c("patientID", "death", "overall_survival_months")]
+         colnames(pat) = c("patient", "death", "OAST")
+         md = read.csv("md_expr.csv", as.is = TRUE)
+         colnames(md) <- gsub("X", "", colnames(md))
+         m2 = read.csv("MDACC_272_Lung_Tumors.csv", as.is = T)[,
+             1:4]
+         mex = merge(m2, md[, -1], by.x = "Probe.Name", by.y = "Illumina.ID")
+         gsig = subset(mex, Symbol %in% gene_names)
+         exp = gsig[, c("Symbol", pat$patient)]
+     }
+     if (dataset == "Consortium") {
+         clin = read.csv("clinical_data12092011Kevin.csv", as.is = T)
+         sur5 = subset(clin, DataSet == "Dataset_5")
+         sur5 = sur5[, -c(2:4, 6)]
+         sur5$stage <- NA
+         sur5$stage[grep("T1", sur5$StageTNM)] <- 1
+         sur5$stage[grep("T2|T3|T4", sur5$StageTNM)] <- 2
+         missing <- sur5[which(is.na(sur5$OS.death) | is.na(sur5$OAST)),
+             c("PatientID")]
+         pat <- subset(sur5, !PatientID %in% missing)
+         pat = pat[, c("PatientID", "OS.death", "OAST")]
+         colnames(pat) = c("patient", "death", "OAST")
+         d5 = read.csv("data5expr.csv", as.is = T)
```

```
+         u133 = read.csv("Affymetrix_U133_from_Luc.csv", as.is = T)
+         d5a = merge(u133, d5, by.x = "Affy.ID", by.y = "X")
+         d5sub = subset(d5a, Symbol %in% gene_names)
+         col.names = colnames(d5sub)
+         col.names = col.names[4:length(col.names)]
+         col.names = col.names[order(col.names)]
+         d5sub = d5sub[, c(colnames(d5sub)[1:3], col.names)]
+         exp = d5sub[, !colnames(d5sub) %in% missing]
+         exp = exp[, 3:dim(exp)[2]]
+     }
+     if (dataset == "Tomida") {
+         load("TomidaGSE13213expr_os.RData")
+         tomida_expr = data$expr
+         tomida_expr = tomida_expr[tomida_expr$Gene.Symbol %in%
+             gene_names, ]
+         retain = c()
+         for (i in 1:dim(tomida_expr)[1]) {
+             if (all(!is.nan(as.matrix(tomida_expr[i, -1])))) {
+                 retain = c(retain, i)
+             }
+         }
+         exp = tomida_expr[retain, ]
+         colnames(exp)[1] = "Symbol"
+         pat = data$clin
+         pat = pat[, c("UniqueID", "death", "overall_survival_months")]
+         colnames(pat) = c("patient", "death", "OAST")
+         pat$patient = as.vector(pat$patient)
+     }
+     result = list(pat, exp)
+     names(result) = c("pat", "exp")
+     result
+ }
```

Define the function to draw KM-plot. The p value calculation is slightly differnt from the one used in the manuscript.

```
> survival_plot = function(gene_names, datasets, censor) {
+     library(superpc)
+     library(affy)
+     library(preprocessCore)
+     x = read.datasets(datasets[1])
+     y = read.datasets(datasets[2])
+     xmean = aggregate(x$exp[, -1], by = list(x$exp$Symbol), mean)
+     ymean = aggregate(y$exp[, -1], by = list(y$exp$Symbol), mean)
+     xymean = merge(xmean, ymean, by = "Group.1")
+     newd = xymean[, ]
```

```
+     newd[, -1] <- normalize.quantiles(as.matrix(newd[, -1]))
+     x$exp <- newd[, x$pat$patient]
+     y$exp <- newd[, y$pat$patient]
+     pv.expr <- function(x, digits = 1) {
+         if (!x)
+             return(0)
+         exponent <- floor(log10(x))
+         base <- round(x/10^exponent, digits)
+         ifelse(x > 1e-06, paste("p = ", base * (10^exponent),
+             sep = ""), paste("p = ", base, "E", exponent, sep = ""))
+     }
+     x$pat$OAST_censored = x$pat$OAST
+     x$pat$death_censored = x$pat$death
+     x$pat[x$pat$OAST > censor, "death_censored"] = 0
+     x$pat[x$pat$OAST > censor, "OAST_censored"] = censor
+     y$pat$OAST_censored = y$pat$OAST
+     y$pat$death_censored = y$pat$death
+     y$pat[y$pat$OAST > censor, "death_censored"] = 0
+     y$pat[y$pat$OAST > censor, "OAST_censored"] = censor
+     data.train <- NULL
+     data.train <- list(x = x$exp, y = x$pat$OAST, censoring.status = x$pat$death,
+         featurenames = newd$Group.1)
+     train.obj <- NULL
+     train.obj <- superpc.train(data.train, type = "survival")
+     data.test <- NULL
+     data.test <- list(x = y$exp, y = y$pat$OAST_censored, censoring.status = y$pat$death_c
+         featurenames = NULL)
+     risk <- NULL
+     risk <- superpc.predict(train.obj, data.train, data.test,
+         threshold = 1, prediction.type = "continuous")$v.pred.1df
+     write.table(file = paste("~/iproject/test/", datasets[1],
+         "_to_", datasets[2], ".txt", sep = ""), matrix(data = risk,
+         ncol = 1, dimnames = list(names(risk), "risk")))
+     surv.fit <- survfit(Surv(y$pat$OAST_censored, y$pat$death_censored) ~
+         risk > median(risk))
+     logrank <- survdiff(Surv(y$pat$OAST_censored, y$pat$death_censored) ~
+         risk > median(risk))
+     pv <- pchisq(logrank$chisq, 1, lower.tail = F)
+     par(mar = c(4, 4, 4, 4), mfrow = c(1, 1))
+     plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Month",
+         ylab = "Survival", mark = 20, cex.lab = 1.5, lwd = 2,
+         main = paste(datasets[1], "to", datasets[2]))
+     text(40, 0.2, pv.expr(pv), cex = 1.5)
+     data.train <- NULL
+     data.train <- list(x = y$exp, y = y$pat$OAST, censoring.status = y$pat$death,
+         featurenames = newd$Group.1)
```

```
+     train.obj <- NULL
+     train.obj <- superpc.train(data.train, type = "survival")
+     data.test <- NULL
+     data.test <- list(x = x$exp, y = x$pat$OAST_censored, censoring.status = x$pat$death_c
+         featurenames = NULL)
+     risk <- NULL
+     risk <- superpc.predict(train.obj, data.train, data.test,
+         threshold = 1, prediction.type = "continuous")$v.pred.1df
+     write.table(file = paste("~/iproject/test/", datasets[2],
+         "_to_", datasets[1], ".txt", sep = ""), matrix(data = risk,
+         ncol = 1, dimnames = list(names(risk), "risk")))
+     surv.fit <- survfit(Surv(x$pat$OAST_censored, x$pat$death_censored) ~
+         risk > median(risk))
+     logrank <- survdiff(Surv(x$pat$OAST_censored, x$pat$death_censored) ~
+         risk > median(risk))
+     pv <- pchisq(logrank$chisq, 1, lower.tail = F)
+     par(mar = c(4, 4, 4, 4))
+     plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Month",
+         ylab = "Survival", mark = 20, cex.lab = 1.5, lwd = 2,
+         main = paste(datasets[2], "to", datasets[1]))
+     text(40, 0.2, pv.expr(pv), cex = 1.5)
+ }
```

Some example commands to run the survival analysis on Alex's gene signature.

```
> setwd("~/projects/ASCL1/data/ASCL1/expression/survival")
> gene_names = read.table("Alex_72.txt", stringsAsFactors = F)[,
+     1]
> censor = 60
> survival_plot(gene_names, c("Tomida", "mda"), censor)
> survival_plot(gene_names, c("Consortium", "mda"), censor)
> survival_plot(gene_names, c("Consortium", "Tomida"), censor)
```

Supplementary Materials and Methods

Cell Lines

All lung cancer cell lines used in this study were obtained from the Hamon Cancer Center Collection (University of Texas Southwestern Medical Center). Cancer cells were maintained in RPMI-1640 (Life Technologies Inc.) supplemented with 5% or 10% fetal calf serum (FCS) without antibiotics at 37°C in a humidified atmosphere containing 5% CO2 and 95% air. All cell lines have been DNA fingerprinted using the PowerPlex 1.2 kit (Promega) and mycoplasma tested by e-Myco kit (Boca Scientific). Cells were treated with ABT-263 (Selleck) or DMSO control for up to 72 hours.

Quantitative RT-PCR

cDNA was generated with an iScript cDNA synthesis kit (BioRad). Gene specific Taq-Man probes (Applied Biosystems) were utilized for quantitative analyses of mRNA transcript levels. The GAPDH gene was employed as an internal reference to normalize input cDNA. PCR reactions were run using the ABI 7300 Real-time PCR System and analyzed with the included software (Applied Biosystems). The comparative C_T method was used to calculate relative mRNA expression levels.

Western Blot Analysis

Cellular proteins were separated by 10% SDS/polyacrylamide gel electrophoresis and electrotransferred to nitrocellulose membranes (Millipore). The membrane was blocked for 1 hour at room temperature (RT) then incubated with a primary antibody overnight at 4°C, followed by incubation with a horseradish peroxidase-conjugated secondary

antibody (Cell Signaling) for 2 hours at RT. Proteins were detected by enhanced chemiluminescence (Thermo Scientific). Primary antibodies against ASCL1 (BD Biosciences), BCL2 (Cell Signaling), PARP and Cleaved PARP (Cell Signaling), Cleaved Caspase 3 (Cell Signaling), RET (Cell Signaling), ALDH1A1 (Cell Signaling), and Hsp90 (Cell Signaling) were used in the study.

Microarray Analysis

Total RNA from cell lines was isolated using RNEasy kit (Qiagen). Gene expression profiling on each sample was performed using Illumina HumanWG-6 V3 BeadArrays (for the 206 lung cell lines GSE32036). Bead-level data were obtained and pre-processed using the R package mbcb for background correction and probe summarization. Pre-processed data were then quartile-normalized and log-transformed for class comparison and unsupervised clustering analysis.

Transient siRNA Transfections

Lung cancer cell lines were optimized for transfection conditions in 6-well and 96-well plates by monitoring lipid content and cell number, and measuring the proliferative differences between scramble oligo control (Qiagen) and toxic control (Qiagen). For 6-well experiments, 3-5 µL RNAiMAX (Invitrogen) was added to 500 uL serum-free RPMI-1640 and incubated at room temperature for 5 minutes. 20 nM siRNA was mixed, plated dropwise in 6-well plates, and complexed for 20 minutes. $200x10^5$ cells were added on top of the mixture, and incubated at 37°C for 72 hours prior to analysis. For 96-well experiments, 0.2-0.4 uL RNAiMAX was added to 10 uL RNAiMAX and incubated for 5 minutes at RT. 20 nM siRNA was added to the lipid mixture and then added to each

well. $2x10^3$ cells were added in 90 uL RPMI supplemented with 5% or 10% FBS and incubated at 37°C for 5 days prior to proliferation analysis by MTS. siRNAs were purchased from Qiagen, including siASCL1-1,-2,-3 (SI00062573, SI00062580, SI00062587) , siBCL2 (SI00299397), siRET (SI02224985), siSOX2, siTTF1, siLUC, and siSCR.

Cell Cycle Analysis

1×10^6 cells suspended in 1 mL PBS were added drop wise into 2.5 mL of cold ethanol. After overnight incubation at -20°C, cells were resuspended and incubated in 500 µl staining solution (0.05% Triton X-100 in PBS supplemented with 50 µg RNase A (Sigma) and 50 µg/mL PI) for 40 min at 37°C. Cell cycle analysis was performed on a FACSCalibur flow cytometer.

MTS Proliferation Assay

Relative cell growth was analyzed by MTS assay. Briefly, 100 uL of cells grown in 96-well plates were mixed with 20 uL MTS assay reagent consisting of tetrazolium compound and phenazine ethosulfate, an electron coupling reagent (Promega). Cells were incubated with MTS mixture until formation of soluble formazan product was observed. Relative absorbance was analyzed by plate reader.

shRNA Stable Expression in Lung Cancer Lines

pGIPZ lentiviral shRNA constructs targeting ASCL1 were purchased from Thermo Scientific. pGIPZ-shNTC served as a negative control. Lentiviruses were packaged in 293T cells. Briefly, 293T cells were cultured in DMEM containing 10% FBS and

transiently transfected with shRNA vector together with pMDG-VSVG and pCMV-ΔR8.91 plasmids using Fugene6 (Roche). After overnight incubation, the viral supernatant was collected, filtered, and used for the transduction of lung cancer cells in the presence of 8 μg/mL polybrene (Sigma-Aldrich). Stable shRNA expressing lung cancer cells were generated after a one-week selection in 1.5 μg/mL puromycin.

Liquid Colony Formation Assays

For anchorage-dependent colony formation, $1x10^3$ cells were plated in 6-well plates. Two weeks later, colonies were stained with 0.5% crystal violet and counted using Image J software (NIH).

Immunohistochemistry

Immunohistochemical (IHC) staining for ASCL1 was performed on tissue microarray and whole section samples as follows: 5 μm-thick formalin-fixed, paraffin-embedded tissue sections were deparaffined, hydrated, and processed in Leica BOND-MAX (Leica Microsystems Inc.). Slides were incubated with the primary antibody (ASCL1 1:25). Staining was developed with chromogen substrate (Leica Microsystems Inc.) and then counterstained with hematoxylin, dehydrated, and mounted. Immunostaining intensity and reactivity were examined by experienced pathologists (J.F. and I.W.) using a light microscope under a 20x magnification objective. ASCL1 nuclear expression was quantified using a 4-value intensity score (0, none; 1, weak; 2, moderate; and 3, strong) and the percentage (0%–100%) of the extent of reactivity. A final expression score was obtained by multiplying the intensity and reactivity extension values (range, 0–300).

***In Vivo* Tumor Xenograft Experiments**

In vivo efficacy of ABT-263 was evaluated through xenografts established from subcutaneous injection of NCI-H1993 and NCI-H1755 cells to the flank of female 5- to 6-week-old NOD/SCID mice. Each mouse was injected with $1x10^6$ viable cells in 0.2 mL of PBS and monitored every 2–3 days for tumor formation. Tumor size was assessed with digital calipers; tumor volume was taken to be equal to the width × length2 × π/6. Once subcutaneous tumor reached approximately 250 mm^3 mice were administered 100 mg/kg ABT-263 or vehicle control (i.p., daily for 14 days) at which point mice were sacrificed and subcutaneous tumors were harvested for analysis. ABT-263 was dissolved in propylene glycol, Tween-80, and D_5W (pH 1.0). The mixture was sonicated and pH adjusted to ~4. All animal care was in accord with institutional guidelines and approved IACUC protocols.

Chromatin Immunoprecipitation, Sequence Library Preparation, and Alignment

10 million lung cancer cells were prepared for chromatin immunoprecipitation (ChIP) by washing twice with cold PBS followed by trypsinization. The cell lines utilized for ChIP were the following: H1755, HCC4018 (NE-NSCLC), H128, H1184, H2107 (SCLC), and control cell lines H524 and H526 (ASCL1(-) SCLC). Nuclei were liberated from cells by dounce homogenization and then fixed in 1% formaldehyde for 10 minutes at room temperature. Fixation was terminated by adding glycine to a final concentration of 0.125M. Chromatin was sheared by using a Diagenode Bioruptor for 30 minutes on high power with 30s:30s on:off cycles. 100 µg chromatin was immunoprecipitated with 5 µg affinity-purified mouse anti-ASCL1 antibody (BD Biosciences) followed by anti-mouse Dyna beads (Invitrogen). The immunoprecipitated chromatin was then purified with the Qiagen PCR Clean-up kit.

Prior to sequencing, ChIP quality was determined by qRT-PCR for known targets DLL1 and DLL3 as well as negative control regions. ChIP-Seq libraries were prepared using the NEBNext ChIP-Seq Library kit. Indexing primers and adapters were obtained from Illumina. Single-end sequencing of 50 bp was conducted for all samples on the Illumina High-Seq 2000 sequencer. The DNA sequencing data produced following ChIP and library preparation were aligned using Bowtie (35). The parameters for running Bowtie are "-S -n 2 -e 70 -l 20 -m 3 --time -p 12 --chunkmbs 512." The reference genome is HG19. Replicates were mapped individually and pooled together.

Peak Calling Using Model-Based Analysis for ChIP-Seq and DNA Motif Analysis

ChIP-Seq peaks were called using Model-Based Analysis for ChIP-Seq (MACS) software, version 1.4.0rc (36). All reads that were mapped to more than one genomic region were removed in order to reduce ambiguity. Additionally, only one unique copy of each read was retained to prevent against PCR bias. Reads from control ASCL1(-) cell lines H524 and H526 were pooled prior to comparison with ASCL1(+) cell line reads. Peak calling was performed using default parameters in MACS. The cutoff for tag reads needed to retain a peak in each cell line varied from 11 to 19. Cutoff values were manually chosen based on visual inspection of ChIP-Seq peaks in the UCSC Genome Browser. Peaks appearing in ASCL1(-) control samples were subtracted from ASCL1(+) samples.

After peak calling was performed in MACS on each cell line, a hierarchical clustering algorithm with complete linkage is used to identify consensus peaks in the ASCL1(+) cell lines. A maximum distance of 300 bp between peak summits appearing in different

samples is allowed for consideration of consensus peaks. For clusters of consensus peaks, a new summit is calculated from summits of member peaks weighted by fold change. DNA motif analysis was performed using Heterogeometric Optimization of Motif EnRichment (HOMER) (37). The parameters used for HOMER are "-S 15 -bits -size -50,50 -len 5,6,7,8,9,10 –keepFiles." The"-50,50" parameter informs HOMER to search for motifs within a 100 bp window centered around the summit of consensus peaks.

Gene Associations using Genomic Region Enrichment Annotation Tool

Consensus peaks identified using MACS were tabulated as 70 bp reads in a .BED file and uploaded to the Genomic Region Enrichment Annotation Tool (GREAT) server in order to correlate genomic peak location with genes (18). Default parameters for gene association were used. GREAT defines a basal regulatory region for a gene within 5 kb upstream of the transcriptional start site (TSS) or 1 kb downstream and defines an extended regulatory region that exists within 1000 kb both up and downstream of the TSS. These rules were utilized to assign gene associations to the consensus peaks obtained from hierarchical clustering of peaks identified via MACS analysis.

Correlation of Associated Genes with Microarray Expression Data

Microarray expression data from the Minna lab in conjunction with ChIP-Seq gene-association data was utilized to find likely transcriptional targets of ASCL1. Only those cell lines utilized for ChIP-Seq (H1755, HCC4018, H128, H1184, and H2107 and controls H524 and H526) were used to compare gene expression of ASCL1 targets. Microarray expression analysis was used to determine $\log_2$ ratio differences in

transcripts between ASCL1(+) samples and the ASCL1(-) control lines. Comparison of gene expression differences of the 1330 ASCL1-associated genes between H1755, HCC4018, H128, H1184, H2107 versus H524 and H526 resulted in a list of 72 ChIP-Seq target genes specifically up-regulated in ASCL1(+) samples.

Survival analysis

The prognostic performance of the 72 gene set was tested on three independent mRNA expression datasets: 442 primary lung adenocarcinomas comprising the National Cancer Institute Director's Challenge Consortium study (Consortium dataset (19)). the Tomida dataset consisting of 119 lung adenocarcinomas (GSE13213) (20), and 209 primary lung adenocarcinomas and squamous cell carcinomas from the SPORE dataset (GSE41271) (21). Overall survival time was defined as the time from the date of surgery to death or last follow-up contact. The prediction model was built from the training set by Supervised Principal Component analysis and then validated in the testing set. The Supervised Principal Component analysis was implemented using superPC R package with all default parameters. The testing set samples were then divided into two equal-sized risk groups by the median of the predicted risk scores. Survival curves were estimated by the Kaplan-Meier method and compared according to log-rank test.