

# xseq – Assessing Functional Impact on Gene Expression of Mutations in Cancer

Jiarui Ding, Sohrab Shah

2015-09-10

## Introduction

The `xseq` model specifies how the expression  $Y$  of a group of genes in a patient is influenced by the somatic mutation status of a gene  $g$  in the patient. The main question we address is whether gene  $g$  co-associates with disrupted expression to itself or its connected genes as defined by an influence graph. This concept is motivated by biological hypotheses predicting that some functional mutations will exhibit a “transcriptional shadow”, resulting from a mechanistic impact on the gene expression profile of a tumour. For example, loss-of-function mutations (nonsense mutations, frame-shifting indels, splice-site mutations or homozygous copy number deletions) occurring in tumour suppressor genes like *TP53* can cause loss of expression due to nonsense-mediated mRNA decay or gene dosage effects. In this context, we define a *cis-effect* as a genetic or epigenetic aberration that results in up-regulation or down-regulation of the gene itself. In contrast, some mutations can disrupt the expression of other genes in the same biochemical pathway (*trans-effects*). This class of mutations tends to cast a long transcriptional shadow over many genes across the genome.  $\beta$ -catenin (*CTNNB1*) mutations, which drive constitutive activation of Wnt signalling in several cancer types, are a potent example of mutational impact on gene expression.

## Inputs

The `xseq` model is predicated on the idea that mutations with functional effects on transcription will exhibit measurable signals in mRNA transcripts biochemically related to the mutated gene –thus imposing a transcriptional shadow across part (or all) of a pathway. To infer this property, three key inputs are required for the model: a patient-gene matrix encoding the presence/absence of a mutation (any form of somatic genomic aberrations that can be ascribed to a gene, e.g., SNVs, indels, or copy number alterations); a patient-gene expression matrix encoding continuous value expression data (e.g., from RNASeq or microarrays); and a graph structure encoding whether two genes are known to be functionally related (e.g., obtained through literature, databases, or co-expression data). `xseq` uses a precomputed ‘influence graph’ as a means to incorporate prior gene-gene relationship knowledge into its modelling framework. For analysis of mutation impact *in-cis*, the graph reduces to the simple case where the mutated gene is only connected to itself.

```
library(xseq)
data(mut, expr, cna.call, cna.logr, net)

mut[1:5,1:5]
```

```
##           sample hgnc_symbol entrezgene variant_type chrom
## 1 TCGA-AB-2802-03      TBX15           0      MISSENSE      1
## 2 TCGA-AB-2802-03      TCHHL1           0      MISSENSE      1
## 3 TCGA-AB-2802-03    ANKRD30A           0      MISSENSE     10
## 4 TCGA-AB-2802-03      PTPN11           0      MISSENSE     12
## 5 TCGA-AB-2802-03      EP400           0      SYNONYMOUS    12
```

```
expr[1:5,1:5]
```

```
##          NPM1      RUNX1      KDM6A      FLT3      TP53
## TCGA-AB-3007-03 13.19168 13.99527 10.178914 12.50839 11.13443
## TCGA-AB-2990-03 13.64727 14.02808 10.099283 13.46506 10.98983
## TCGA-AB-2915-03 12.60635 13.45520 11.060389 12.23997 10.98188
## TCGA-AB-2927-03 12.61067 14.48913 10.743187 13.54870 10.89469
## TCGA-AB-3000-03 13.26361 11.88463  9.955398 13.40446 11.56106
```

```
cna.call[1:5,1:5]
```

```
##          NPM1 RUNX1 KDM6A FLT3 TP53
## TCGA-AB-2803-03  0    0    0    0    0
## TCGA-AB-2804-03  0    0    0    0    0
## TCGA-AB-2805-03  0    0    0    0    0
## TCGA-AB-2806-03  0    0    0    0    0
## TCGA-AB-2807-03  0    1    0    0    0
```

```
cna.logr[1:5,1:5]
```

```
##          NPM1      RUNX1      KDM6A      FLT3      TP53
## TCGA-AB-2884-03 -0.0065  0.0056  0.0110 -0.0023 -0.0015
## TCGA-AB-2943-03 -0.0878 -0.0800  0.0026 -0.0527 -0.1136
## TCGA-AB-2938-03  0.0384  0.0004 -0.0044  0.0183  0.9600
## TCGA-AB-2806-03  0.0017 -0.0062 -0.0005  0.0011  0.0194
## TCGA-AB-2826-03  0.0230  0.0123  0.0069  0.0095  0.0091
```

```
net[1:2]
```

```
## $NPM1
##      MDM2      AKT1      HEXIM1      GRK5      SSB      NOP2      HMGA2
##      1      1      1      1      1      1      1
##      PPIID HIST2H2BE      BAALC      AFF1      TET2      THOC7      HOXA7
##      1      1      1      1      1      1      1
##      KPNB1      PC      SMC4      IDH1      KIT      CD34      STAT5B
##      1      1      1      1      1      1      1
##      MN1
##      1
##
## $RUNX1
##      SUV39H1      ETS1      SMAD4      RUNX3      IGFBP3      PIM1      CEBPA      TLE1
##      1      1      1      1      1      1      1      1
##      CCND1      SMAD3      SMARCA4      AR      TCF12      SMAD1      LMO2      CD34
##      1      1      1      1      1      1      1      1
##      NT5DC3      HRAS      ZNF687      CDC73      NOTCH2NL      YTHDF2      PBX1      JAK2
##      1      1      1      1      1      1      1      1
##      FHOD1      ZFPM1      ABL1      ZFP64      GFI1      BAALC
##      1      1      1      1      1      1
```

## Cis-analysis

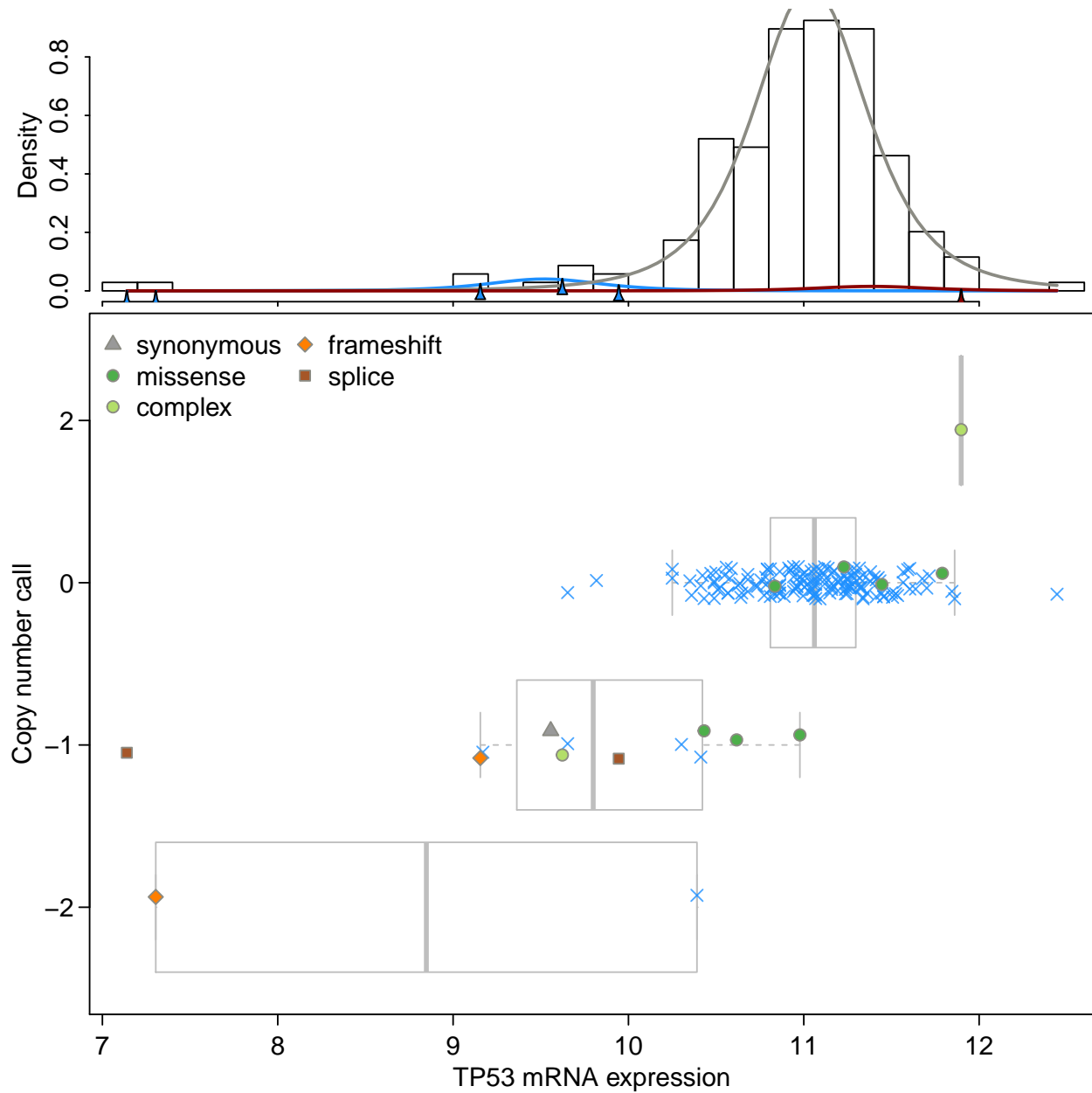
We first analyze the cis-effects of loss-of-function mutations (frameshift, nonsense and splice-site mutations) on gene expression.

```
# Compute whether a gene is expressed in the studied tumour type.
# If the expression data are from microarray, there is not need to compute weights.
weight      = EstimateExpression(expr)

# Impute missing values
expr        = ImputeKnn(expr)
cna.logr    = ImputeKnn(cna.logr)

# Quantile-Normalization
expr.quantile = QuantileNorm(expr)

#####
## Get the conditional distributions P(Y|G)
#
# We first show TP53 mutations, expression, and copy number alterations
tmp = GetExpressionDistribution(expr=expr.quantile, mut=mut, cna.call=cna.call,
                               gene="TP53", show.plot=TRUE)
```



```
expr.dis.quantile = GetExpressionDistribution(expr=expr.quantile, mut=mut)
```

```
#####
## Filtering not expressed genes, and only analyzing loss-of-function
## Mutations
##
id = weight[mut[, "hgnc_symbol"]] >= 0.8 &
      (mut[, "variant_type"] %in% c("FRAMESHIFT", "NONSENSE", "SPLICE"))
id = id & !is.na(id)
mut.filt = mut[id, ]
```

```
#####
init = SetXseqPrior(expr.dis = expr.dis.quantile,
```

```

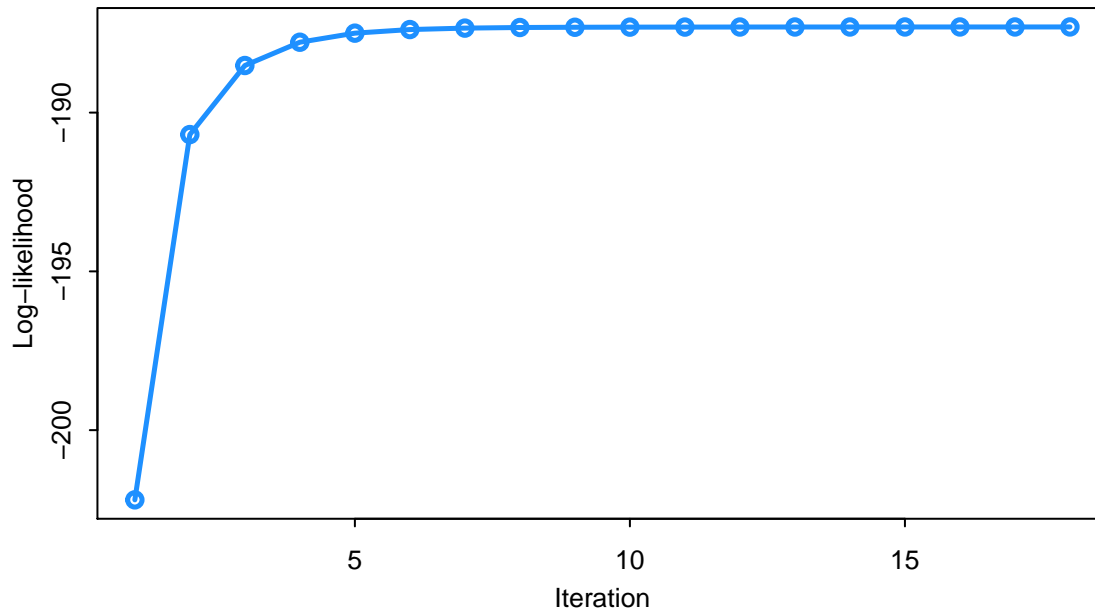
mut      = mut.filt,
mut.type = "loss",
cis      = TRUE)

# Parameter constraints in EM-iterations
constraint = list(equal.fg=FALSE)

model.cis = InitXseqModel(mut      = mut.filt,
                          expr      = expr.quantile,
                          expr.dis  = expr.dis.quantile,
                          cpd       = init$cpd,
                          cis       = TRUE,
                          prior     = init$prior)

model.cis.em = LearnXseqParameter(model      = model.cis,
                                  constraint = constraint,
                                  iter.max   = 50,
                                  threshold  = 1e-6)

```



```

xseq.pred = ConvertXseqOutput(model.cis.em$posterior)
xseq.pred[1:20,]

```

##	sample	hgnc_symbol	P(F)	P(D)
## 67	TCGA-AB-2820-03	TP53	0.9639755	0.9905104
## 68	TCGA-AB-2857-03	TP53	0.9613231	0.9905104
## 69	TCGA-AB-2860-03	TP53	0.9818841	0.9905104
## 70	TCGA-AB-2868-03	TP53	0.9648281	0.9905104
## 71	TCGA-AB-2908-03	TP53	0.9796830	0.9905104
## 72	TCGA-AB-2938-03	TP53	0.7893094	0.9905104
## 113	TCGA-AB-2871-03	STAG2	0.9749255	0.9846371
## 114	TCGA-AB-2913-03	STAG2	0.7286295	0.9846371
## 115	TCGA-AB-2964-03	STAG2	0.9779552	0.9846371
## 116	TCGA-AB-2972-03	STAG2	0.9769733	0.9846371

```

## 117 TCGA-AB-2978-03      STAG2 0.9789665 0.9846371
## 62  TCGA-AB-2818-03      RAD21 0.8466841 0.9823520
## 63  TCGA-AB-2886-03      RAD21 0.9700700 0.9823520
## 64  TCGA-AB-2967-03      RAD21 0.9708465 0.9823520
## 65  TCGA-AB-2975-03      RAD21 0.9654264 0.9823520
## 66  TCGA-AB-2986-03      RAD21 0.9684474 0.9823520
## 118 TCGA-AB-2900-03      SMC1A 0.8430609 0.6738602
## 9   TCGA-AB-2807-03      POLR2E 0.7352240 0.5944941
## 108 TCGA-AB-2851-03      SMC3 0.7756835 0.5110214
## 109 TCGA-AB-2950-03      SMC3 0.4095276 0.5110214

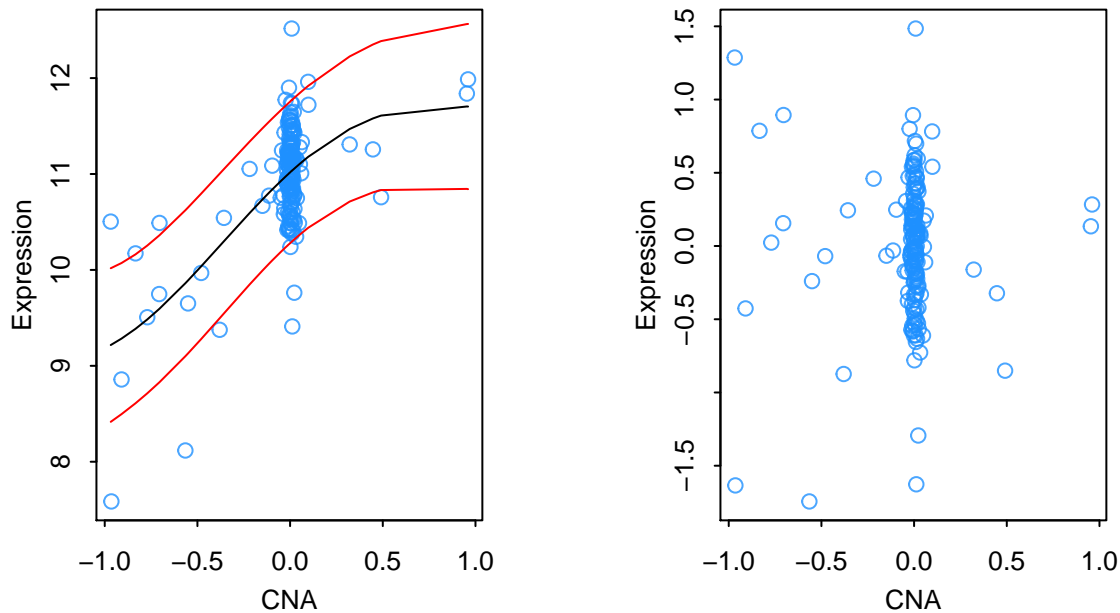
```

## Trans-analysis

```

#-----
## Remove the cis-effects of copy number alterations on gene expression
#
# We show an example: PTEN copy number alterations and expression in AML
tmp = NormExpr(cna.logr=cna.logr, expr=expr, gene="TP53", show.plot=TRUE)

```



```

expr.norm = NormExpr(cna.logr=cna.logr, expr=expr)
expr.norm.quantile = QuantileNorm(expr.norm)

#-----
## Get the conditional distributions P(Y|G),
#
expr.dis.norm.quantile = GetExpressionDistribution(expr=expr.norm.quantile,
                                                  mut=mut)

#-----
##

```

```

## Filtering not expressed genes
##

id = weight[mut[, "hgnc_symbol"]] >= 0.8
id = id & !is.na(id)
mut.filt = mut[id, ]

#####
# Filter the network
net.filt = FilterNetwork(net=net, weight=weight)

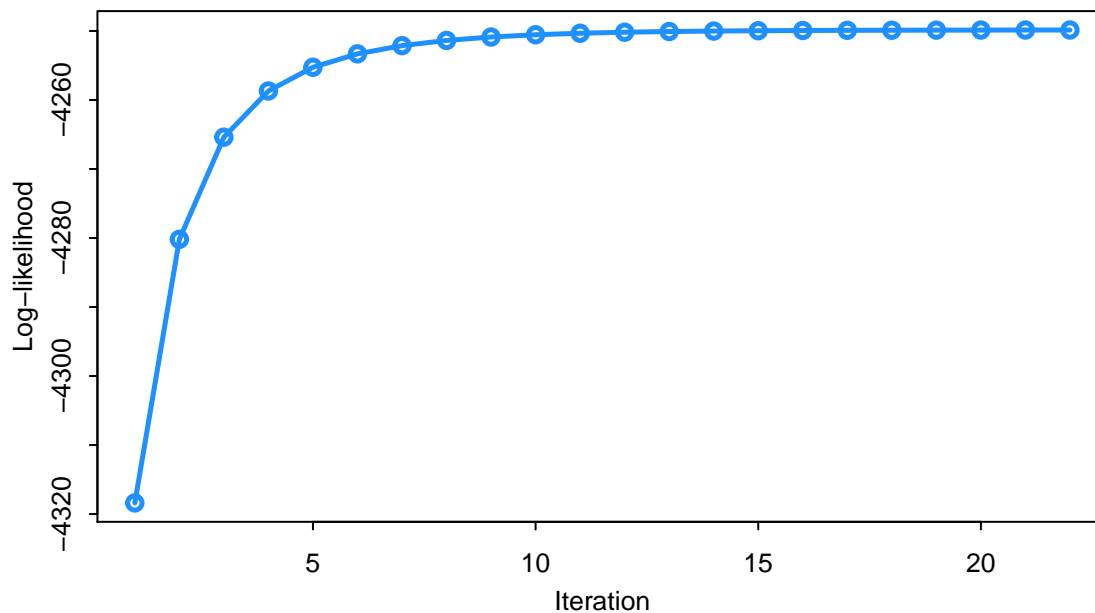
init = SetXseqPrior(expr.dis = expr.dis.norm.quantile,
                    net      = net.filt,
                    mut      = mut.filt,
                    mut.type = "both",
                    cis       = FALSE)

# parameter constraints in EM-iterations
constraint = list(equal.fg=TRUE, baseline=init$baseline)

model.trans = InitXseqModel(mut      = mut.filt,
                             expr    = expr.norm.quantile,
                             net     = net.filt,
                             expr.dis = expr.dis.norm.quantile,
                             cpd     = init$cpd,
                             cis     = FALSE,
                             prior    = init$prior)

## EM algorithm for parameter estimations
model.trans.em = LearnXseqParameter(model      = model.trans,
                                    constraint = constraint,
                                    iter.max   = 50,
                                    threshold  = 1e-6)

```



```

=====
# Reformat output

xseq.pred = ConvertXseqOutput(model.trans.em$posterior)
xseq.pred[1:20, ]

##           sample hgnc_symbol      P(F) P(D)
## 49 TCGA-AB-2810-03      NPM1 0.7243979  1
## 50 TCGA-AB-2811-03      NPM1 0.8105504  1
## 51 TCGA-AB-2812-03      NPM1 0.9998626  1
## 52 TCGA-AB-2816-03      NPM1 0.8480545  1
## 53 TCGA-AB-2818-03      NPM1 0.9962872  1
## 54 TCGA-AB-2824-03      NPM1 0.9205875  1
## 55 TCGA-AB-2825-03      NPM1 0.9832970  1
## 56 TCGA-AB-2826-03      NPM1 0.8589251  1
## 57 TCGA-AB-2835-03      NPM1 0.7871754  1
## 58 TCGA-AB-2836-03      NPM1 0.5572454  1
## 59 TCGA-AB-2837-03      NPM1 0.9038296  1
## 60 TCGA-AB-2839-03      NPM1 0.9579794  1
## 61 TCGA-AB-2848-03      NPM1 0.9568224  1
## 62 TCGA-AB-2853-03      NPM1 0.9999716  1
## 63 TCGA-AB-2859-03      NPM1 0.5737191  1
## 64 TCGA-AB-2861-03      NPM1 0.9971857  1
## 65 TCGA-AB-2869-03      NPM1 0.9100203  1
## 66 TCGA-AB-2871-03      NPM1 0.6010043  1
## 67 TCGA-AB-2877-03      NPM1 0.9848812  1
## 68 TCGA-AB-2879-03      NPM1 0.9763781  1

# We finally show the dysregulation probabilities of genes connected to TP53
tmp = PlotRegulationHeatmap(gene="TP53", posterior=model.trans.em$posterior, main="in_AML",
                             mut=mut, subtype=list(NULL), key=FALSE, dendrogram="row")

```



# TP53\_in\_AML

