SpatialStats: statistical models of tumour cell signalling

Kieran Campbell
kieran.campbell@dpag.ox.ac.uk

August 28, 2014

Contents

1 Introduction

1.1 Signalling model

This vignette introduces the workflow for elucidating spatial signalling pathways from spatially resolved proteomics data. The proposed model is of the form

$$y_i^{(p)} = \mu_p + \sum_{j \in NN(i)} w_{j|i} \sum_{q \in P} \beta_{qp} x_j^{(q)} + \epsilon_{ip} \tag{1}$$

where $y_i^{(p)}$ is the concentration of protein p in cell i, $NN(i)$ are the nearest neighbour cells of i, $w_{j|i}$ is the relative boundary size of cell j to cell i (that is, the proportion of cell i's boundary made up of cell j)[1], P is the set of all proteins considered, β_{qp} is the effect of nearest neighbour protein q on protein p in the central cell, and $x_j^{(q)}$ is the concentration of protein q in cell j. The eventual quantities of real interest are the $\{\beta\}$, which give a measure of one protein's effect on the other in nearby cells and would eventually elucidate inter-cellular signalling pathways. The simple average can also be used, rather than the average weighted by the relative boundary size. The effect of this is that $w_{j|i} = \frac{1}{N_i}$ if cell i has N_i neighbours.

1.2 LASSO regression

Rather than ordinary least squares regression, *SpatialStats* uses LASSO regression - a form of regularization where the magnitude of each coefficient is penalised. This is motivated on three fronts: (1) it avoids overfitting given the large number of predictor variables, (2) it scales well to $p > n$ probelems of a subset of cells is selected, and (3) it fits well with the assumption that the underlying interaction matrix is sparse, since we don't expect many channels to spatially interact with the others.

[1] Note that in general $w_{j|i} \neq w_{i|j}$

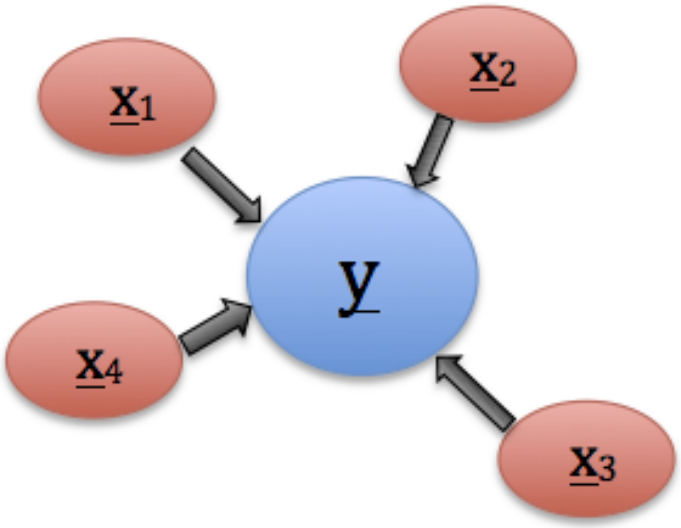

Figure 1: Schematic diagram of the model.

In LASSO regression the objective function becomes

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_i \left(y_i - \sum_j \beta_j x_{ij} \right)^2 + \lambda \sum_j |\beta_j| \tag{2}$$

where the penalisation parameter λ is introduced, which controls the penalty the L^1 norm of the coefficients has on the minimisation. Several packages are available in R to do LASSO regression. Here we use two - *glmnet* and *lars* - depending on the type of significance testing used (see below).

1.3 Significance testing in LASSO

Two significance testing methods in LASSO are used: the multisample splitting method [2], and the recent covariance test statistic [1]. Due to the considerable assumptions underlying our model and noise in the dataset, the final interactions we deem significant are the consensus set - those reported as significant through both methods.

1.3.1 Multi-sample split method

The multisample splitting method is an extension of an earlier method in which the dataset is randomly split into two halves, with feature selection performed on one half and the classical OLS significance testing on the other half using the selected features. The multisample splitting technique extends this so that rather than only splitting the data once the data is randomly split in two multiple times, avoiding results that could be biased due to the arbitrary split of the data.

The main issue that arises from this is how to calculate a single p-value given the multiple splits. First, the Bonferroni correction is applied to each split, so if s features are selected in split b for variable i then the p-value P_i is corrected to $P^{(b)}_{corr,i} = s \cdot P^{(b)}_i$. Note that if variable i is not selected as a feature in split b then the corresponding p-value is simply set to 1. Then if the function

$$Q_i(\gamma) = \min \left(\text{empirical } \gamma\text{−quantile} \left\{ P^{(b)}_{corr,i};\ \ b = 1, \ldots, B \right\}, 1 \right) \tag{3}$$

is defined it can be shown that the corrected p-value for variable i is given by

$$P_j = \min \left((1 - log(\gamma_{min})) \inf_{\gamma \in (\gamma_{min}, 1)} Q_j(\gamma), 1 \right). \tag{4}$$

1.3.2 Covariance test statistic

The covariance test statistic is a very recent advance in significance testing in the context of LASSO and perhaps the most similar to a traditional statistic. As the penalisation parameter λ is decreased, more and more coefficients are re-introduced. The covariance test statistic T_k relies on the difference to the residual sum of squares depending on whether or not a given predictor is included in the model. The authors went on to show that this statistic is asymptotically exponential, meaning a p-value can be found. While a fully mathematical treatment can be found in the original paper [1], here we use the `R` package `covTest` provided by the authors.

1.4 Multiple testing corrections

If we have p channels then the resulting interaction matrix has p^2 entries - in other words we perform p^2 significance tests, which carries a serious risk of calling something significant when it's not. As a result stringent multiple comparison corrections are required. For a given response variable we Bonferroni correct the p-values for the predictors, then for a given predictor we Holm-Bonferroni correct for the response variables (see figure 2).

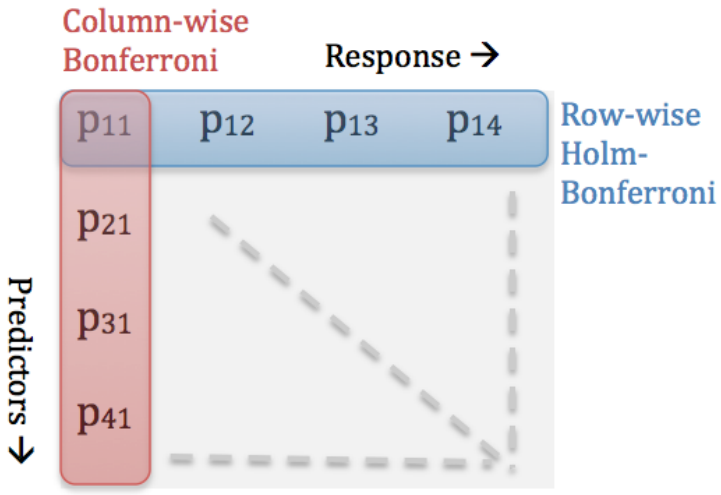

Figure 2: Schematic diagram of multiple testing corrections used.

2 Workflow

By default the *SPExp* object `SPE` is included in the *SpatialPRo* and loaded by default. We can renormalize using all the default settings:

```
library(lars)
library(covTest)

library(SpatialPRo)
library(SpatialStats)

set.seed(123)

SPE@spdata <- lapply(SPlist(SPE), normalizeSP)
```

Since there were very few immune cells in the analysis, it is helpful to remove the immune marks (CD*) before continuing:

```
spe.list <- list()
immune.ind <- c(5,7,8,10,19,28)
for(i in 1:length(SPE)) spe.list[[i]] <-  SPE[[i]][,-immune.ind]

spe.noimmune <- SPExperiment(getDir(SPE), files(SPE), spe.list)
```

We then wish to find the tumour cell class in each sample. This is done by finding the class with the lower median vimentin:

```
tumour.classes <- findIDs(spe.noimmune, "Vimentin", "lower", "median")
```

We then bind all the data together using `BindSPE` and carefulyl construct sample-dependent factors using `ConstructSampleFactors`:

```
XY <- BindSPE(spe.noimmune, choose.class = tumour.classes)
X <- XY$X
Y <- XY$Y

factors <- ConstructSampleFactors(XY, IDs(SPE))
X <- cbind(X, factors)
```

If we wanted to randomise the cells to to a sanity check on the analysis, we would call

```
X <- X[sample(nrow(X)),]
```

We then perform the regression methods. By default `weighted=FALSE`, so we do not weight the regression by relative boundary size. The multi-sample split method is performed using `GeneralLassoSig`:

```
npred <- ncol(X)
include <- npred:(npred - ncol(factors) + 1)

## multisample split results
multisplit.res <- GeneralLassoSig(Y,X,B=100, s="usefixed",fixedP = 5, include=include)
```

And the covariance test statistic results using the `covTest` function:

```
lar <- apply(Y, 2, function(y) lars(X, y, "lasso", normalize=FALSE))
cvtests <- lapply(1:length(lar), function(i) covTest(lar[[i]], X, Y[,i]))
```

The `AdjustCovtests` function takes `cvtests` and turns it into a p by p matrix of p-values where entry (i, j) corresponds to the p-value evidence for interaction i to j.

```
cv.results <- AdjustCovtests(cvtests, ncol(X))
```

It is the necessary to perform the row-wise Holm-Bonferroni multiple

```
alpha <- 0.05

cv.results <- apply(cv.results, 1, p.adjust, method="holm")
cv.results <- t(cv.results)

multisplit.res <- apply(multisplit.res, 1, p.adjust, method="holm")
multisplit.res <- t(multisplit.res)
```

Finally we can perform significance testing at level α:

```
covtest.res <- which(cv.results < alpha, arr.ind=TRUE)
multi.res <- which(multisplit.res < alpha, arr.ind=TRUE)
rownames(multi.res) <- NULL

all.res <- list(covtest.res, multi.res)
```

The `findOverlap` function is used to find overlapping pathways using the multisample-split and covariance test statistic methods. We perform this for pathways of the same componenet (i.e. $i \rightarrow i$) and between different ones (i.e. $i \rightarrow j \neq i$).

```
pathways.diff <- findOverlap(all.res, remove="different")

pathways.same <- findOverlap(all.res, remove="same")

pathways.diff

##
## [1,]

pathways.same

##       [,1]                [,2]
##  [1,] "ER(La139)D"        "ER(La139)D"
##  [2,] "PR(Pr141)D"        "PR(Pr141)D"
##  [3,] "pSHP2(Nd142)D"     "pSHP2(Nd142)D"
##  [4,] "p53(Nd143)D"       "p53(Nd143)D"
##  [5,] "CD31(Nd144)D"      "CD31(Nd144)D"
##  [6,] "Twist(Nd145)D"     "Twist(Nd145)D"
##  [7,] "CD68(Nd146)D"      "CD68(Nd146)D"
##  [8,] "CD3(Sm147)D"       "CD3(Sm147)D"
##  [9,] "Slug(Sm148)D"      "Slug(Sm148)D"
## [10,] "CD20(Sm149)D"      "CD20(Sm149)D"
## [11,] "c-myc(Nd150)D"     "c-myc(Nd150)D"
## [12,] "Her2(Eu151)D"      "Her2(Eu151)D"
## [13,] "pAMPK(Sm152)D"     "pAMPK(Sm152)D"
## [14,] "pAkt(Eu153)D"      "pAkt(Eu153)D"
## [15,] "p44(Gd154)D"       "p44(Gd154)D"
## [16,] "NFkB(Gd156)D"      "NFkB(Gd156)D"
## [17,] "pGSK3(Gd158)D"     "pGSK3(Gd158)D"
## [18,] "pBad(Tb159)D"      "pBad(Tb159)D"
## [19,] "CD44(Dy160)D"      "CD44(Dy160)D"
## [20,] "Vimentin(Dy162)D"  "Vimentin(Dy162)D"
## [21,] "CK7(Dy164)D"       "CK7(Dy164)D"
## [22,] "bcatenin(Ho165)D"  "bcatenin(Ho165)D"
## [23,] "CAH9(Er166)D"      "CAH9(Er166)D"
## [24,] "ECadherin(Er167)D" "ECadherin(Er167)D"
## [25,] "Ki67(Er168)D"      "Ki67(Er168)D"
## [26,] "EGFR(Tm169)D"      "EGFR(Tm169)D"
```

References

[1] Richard Lockhart, Jonathan Taylor, Ryan J Tibshirani, Robert Tibshirani, et al. A significance test for the lasso. *The Annals of Statistics*, 42(2):413–468, 2014.

[2] Nicolai Meinshausen, Lukas Meier, and Peter Bühlmann. P-values for high-dimensional regression. *Journal of the American Statistical Association*, 104(488), 2009.