Extracting mutational signatures using LASSO

Avantika Lal, Daniele Ramazzotti

August 3, 2017

1 Mutational signatures

Point mutations occurring in a genome can be divided into 96 categories based on the base being mutated, the base it is mutated into and its two flanking bases. Therefore, for any patient, it is possible to represent all the point mutations occurring in that patient's tumor as a vector $\mathbf{m}$ of length 96; $\mathbf{m} = [m_{1...}m_{96}]$, where each element m_j represents the count of mutations of category j in the patient.

A mutational signature represents the pattern of mutations produced by a mutagen or mutagenic process inside the cell. Each signature can be represented by a vector $\mathbf{s}$ of length 96; $\mathbf{s} = [s_{1...}s_{96}]$ where each element s_j represents the probability that this particular mutagenic process generates a mutation of category j. Since these are probabilities, $\sum\limits_{j=1}^{96} s_j = 1$.

A patient's tumor genome can be exposed to multiple mutagenic processes, at different intensities. Therefore the vector of mutations in a single patient's tumor can be considered to be a weighted sum of K mutational signatures (K is an unknown number).

$$\mathbf{m} = \sum_{i=1}^{K} \alpha_i \mathbf{s_i} \tag{1}$$

Where α_i is the exposure of the patient to the mutagenic process with signature $\mathbf{s_i}$.

Our goal here is to extract the mutational signatures that best explain the mutation counts of a large number of patients.

2 Matrix Representation

When dealing with multiple patients, we can represent their mutation counts in matrix form. $M_{n \times J}$ is the matrix of counts, where each row represents the i^{th} patient and each column represents the j^{th} category. M_{ij} is the number of mutations of the j^{th} category in the i^{th} patient.

n = Number of patients

J = number of mutation categories (96)
K = number of signatures
Alexandrov et al. have represented M as follows:

$$M = \alpha\beta \tag{2}$$

Where
$\alpha_{n\times K}$ is the weights matrix. α_{ij} is the weight for the j^{th} signature in the i^{th} patient.

$\beta_{K\times J}$ is the signature matrix, where each row represents a signature. β_{ij} is the proportion of mutations in the i^{th} signature that fall into the j^{th} category.

They then use NNMF (non-negative matrix factorization) to solve equation (2) for α and β.

3 Improvements to the method of Alexandrov et al.

We propose two improvements to this method:

1. We introduce a null model based on genome frequencies of trinucleotides. This represents the pattern of mutations that would be expected by random chance, without any specific mutational process. Effectively, it is a signature of randomness. We represent this by a vector $\mathbf{m_0}$ of length J.

2. We also introduce a sparsity constraint. While signatures are J-dimensional vectors, each mutagenic process in the cell is expected to affect only a few specific trinucleotides. Therefore, we expect the matrix β to be sparse.

Based on these, we now represent the matrix M as:

$$M = \alpha_{\mathbf{0}}^{\intercal}\mathbf{m_0} + \alpha\beta + \lambda\|\beta\| \tag{3}$$

Where $\alpha_{\mathbf{0}}$ is a vector of weights of length n.

4 Solving

Given M, $\mathbf{m_0}$ and λ, we can obtain the maximum likelihood values of α and β using a three-step approach. The first two steps are performed in an iterative EM (expectation maximization) manner to maximize the fit and enhancing sparsity in the resulting signatures.

1. Fit $M = \alpha_{\mathbf{0}}^{\intercal}\mathbf{m_0} + \alpha\beta$ using nnls (non-negative least squares)

2. Fit the residual $M - \alpha_{\mathbf{0}}^{\intercal}\mathbf{m_0} = \alpha\beta + \lambda\|\beta\|$ using nnlasso (Non-Negative Lasso). Currently we use the signatures derived from NMF as a starting point for our first two steps.

3. At the end of the EM, normalize each row of β to sum to 1 to get rates.

Here we assume that the null model $\mathbf{m_0}$ is known. If it is unknown, it is also possible to infer both $\mathbf{m_0}$ and $\alpha_\mathbf{0}$ using SVD.

5 Cross-validation to choose λ and K

We can use the following cross-validation strategy to choose optimal values of λ and K:

1. Randomly leave out 10% of the cells in the matrix M to obtain matrix M'.
$$M' = \begin{pmatrix} m_{11} & [] & m_{13} \cdots & m_{1J} \\ [] & m_{22} & m_{23} \cdots & m_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & \cdots & [] & m_{nJ} \end{pmatrix}$$
2. Solve equation (3) using M' in place of M (We have to somehow impute the missing values).
3. Predict the missing values.
4. Compute average prediction error for multiple values of λ and K.
5. Select values of λ and K that minimize prediction error.

6 Accounting for individual copy number variation

Tumors have copy number changes, so the actual trinucleotide frequency is expected to differ slightly from patient to patient. Thus there should ideally be n indipendent null models (one for each patient). If we wish to account for this, we can define the null model M_0 as a n x J matrix.

$$M = \alpha_\mathbf{0}^\intercal \cdot M0 + \alpha\beta + \lambda||\beta|| \tag{4}$$

Where:

$$\alpha_\mathbf{0}^\intercal \circ M0 = (\alpha_i \cdot m_{ij}) = \begin{pmatrix} \alpha_1 \cdot m_{11} & \cdots & \alpha_1 \cdot m_{1J} \\ \vdots & \ddots & \vdots \\ \alpha_n \cdot m_{n1} & \cdots & \alpha_n \cdot m_{nJ} \end{pmatrix}$$