Detecting mutational signatures with confidence

Xiaoqing Huang, Damian Wojtowicz, Teresa M. Przytycka

1. Motivation: The confidence of dectecting the mutational signatures

The terminology mutational signatures, in other words the characteristic patterns of somatic mutations, has been first proposed and inferred by Alexandrov and colleagues(Alexandrov et al. 2013, Nik-Zainal et al. (2016)). By analyzing the mutational context, 96 mutational types in total, which are defined by 5’ and 3’ one nucleotide flanking of the mutated nucleotide, the underlying distinct mutational processes which are operative and active during the tumor progression of the particular patients can be detected and recognized by detecting the mutational signatures. The technology has been applied across many cancer types and 30 signatures with more or less biological understanding have been detected and collected by COSMIC. Not negligibly, as the study on mutational signatures have great positive implications for better understanding of cancer initiation, evolution, and precision cancer treatment, the study on how to decompose a to-be-studied tumor profile into some known signatures is at least as important. With this exact and tailored information, such as what are the intensities of the signatures that are exposed in this particular tumor, and what are the confidences, we can associate them with the corresponding biological mutational processes that are operative in this sample and acquire the insights of the tumor growth, so as the application of personalized treatment would be tremendously improved.

When it comes to decompose a tumor mutational catalogue into a linear combination of some known signature matrix, previous work which adopted NMF techniques has been widely well accepted. New tool called deconstructSigs (dS)(Rosenthal et al. 2016). has also shown its good performance. As we will introduce in methodology section, quadratic programming (QP) is a very stable and efficient algorithm to solve the minimization problem in the same setting, which can reach minimal decomposition error in several seconds. As QP has some restrictions on the known signature matrix, we proposed simulated Annealing (SA) to generalized the application.

Here, we analyzed the biggest cancer whole-genome dataset which contains 560 patients tumor profiles using all these methods respectively, and discrepancies can be detected easily as shown in the figure below.

2. Methododlogy: From point estimation to the distribution

Cancer is a mixture of operative mutational processes, and signatures are the imprints of the particular mutational processes. Specifically, for individual patient, if we can interpret the tumor catalogue in terms of how certain signatures are combined with what intensities of exposures with high power, we will have more confidence in prescribing personalized medicine for cancer treatment and therapy.

2.1 Quadratic programming

Taking advantage of the COSMIC mutational signature matrix P, as columns of P are linearly independent so $\mathrm{P}^{\mathrm{T}}\mathrm{P}$ is positive definite, our minimization problem is equivalent to optimization of a strictly convex quadratic problem. Dual method(Goldfarb and Idnani 1983) is an excellent method to get an efficient and numerically stable solutions to this kind of quadratic problems by utilizing the Cholesky and QR factorizations and updating procedures.

2.2 Simulated Annealing

Another popular optimization algorithms is simulated annealing. Here, SA was generalized according to Tasllis statistics (Tsallis and Stariolo 1996) with a unified technique from classical simulated annealing and fast simulated annealing(Xiang et al. 2013).

The two algorithms both can get the optimal solutions almost exact the same, but have their own strengths and weaknesses, as QP is extremely fast and stable, and SA can be widely used on a not-well-defined objective function. However, one mayor limitation of QP we used here is that the predefined signature matrix has to be positive definite, otherwise the algorithm will fail. While Without loss of generality, SA introduced here sometimes can be slow and take long to converge to the optimal solution.

3. Usage: Analyze the biggest breast cancer whole genome

3.1 Compare four decomposing methods

In the following, four decomposing methods are applied to three selected whole genome tumor catalogues and the similarities and discrpencies are shown. The function findSigExposures which can specify the decomposing method through argument decomposition.method.

```
library(SignatureEstimation)

NMF = read.csv("~/Documents/ICGC/data/NMF_brca.csv", check.names = F, row.names = 1)
deSig = read.csv("~/Documents/ICGC/data/deSig_brca.csv", check.names = F, row.names = 1)
sample_ID_ascending = colnames(NMF)  ## in the order of ascending patient ID
## all 12 signatures
sigsBRCA = c(1, 2, 3, 5, 6, 8, 13, 17, 18, 20, 26, 30)
## subset COSMIC signature matrix
P = signaturesCOSMIC[, sigsBRCA]
## select patients to get figure1A
selected_patients = c("PD13608")
## mutation counts for selected patients
size = c(4105)
names(size) = selected_patients
ymax = c(0.4)
names(ymax) = selected_patients
names(size) = selected_patients
QP = findSigExposures(tumorBRCA[, selected_patients], signaturesCOSMIC[, sigsBRCA],
    decomposition.method = decomposeQP)$exposures
colnames(QP) = selected_patients
SA = findSigExposures(tumorBRCA[, selected_patients], signaturesCOSMIC[, sigsBRCA],
    decomposition.method = decomposeSA)$exposures
colnames(SA) = selected_patients
```

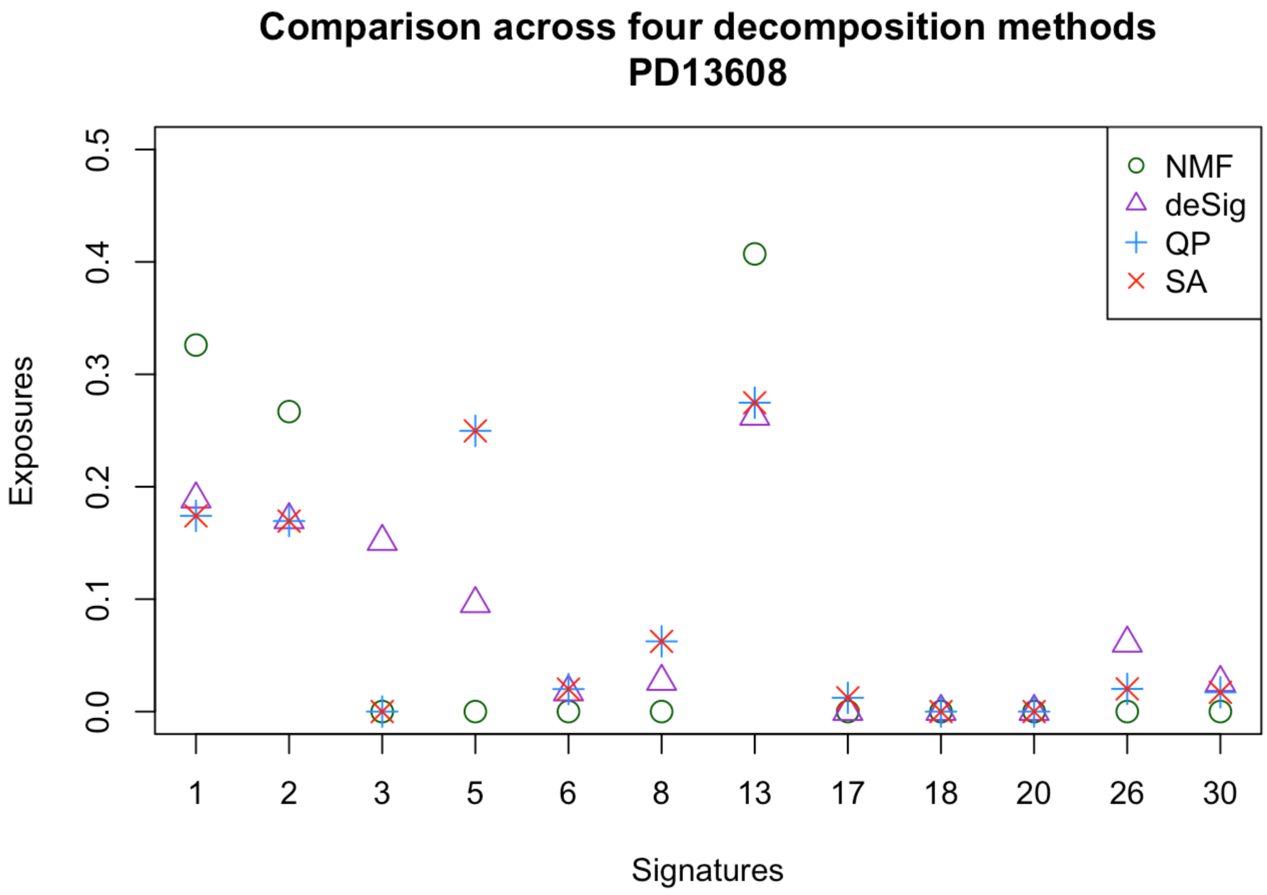
or use code:

```
QP = data.frame(matrix(NA, nrow = ncol(P), ncol = length(selected_patients)))
colnames(QP) = selected_patients
rownames(QP) = paste0("Signature ", sigsBRCA)

SA <- QP

for (sample.i in selected_patients) {
    print(sample.i)
    QP[, sample.i] = decomposeQP(tumorBRCA[, sample.i], P)
    SA[, sample.i] = decomposeSA(tumorBRCA[, sample.i], P)
}
```

```
## generate individual patients
plot(NMF[, selected_patients], xaxt = "n", pch = 1, col = "darkgreen", ylim = c(0,
    0.5), ylab = "Exposures", xlab = "Signatures", main = paste0("Comparison across four
decomposition methods",
    "\n", selected_patients), cex = 1.5)
axis(1, 1:12, labels = as.character(sigsBRCA), las = 1)
points(deSig[, selected_patients], pch = 2, col = "darkorchid", cex = 1.5)
points(QP[, selected_patients], pch = 3, col = "dodgerblue", cex = 1.5)
points(SA[, selected_patients], pch = 4, col = "red", cex = 1.5)
legend("topright", c("NMF", "deSig", "QP", "SA"), col = c("darkgreen", "darkorchid",
    "dodgerblue", "red"), pch = 1:4)
```



The observations suggest that for the same patient, there are some disagreement accorss the methods. This raises a question that which approach we should use when it comes to a particular patient, and how much we can trust the results. Again, this will be very useful for personalized treatment and precision medicine. As mentioned above, NMF is designed mainly for discover signatures that are present in the tumor profile, and dS is a fast approach to determine the exposure intensities, but the decomposing error is not as small as QP or SA, thus QP and SA are preferable approaches to use when calling the signature exposures in term of minimal decomposing error. Nevertheless, the question how much we can trust the results when the tumor catalog is decomposed remains.

3.2 Bootstrap distribution

In order to obtain the confidence of the exposure intensities, bootstrap technique is always a natural solution by establishing bootstrap distributions of the estimates. In this way, randomized re-sampling with replacement was performed on original mutation data to mimic the perturbation of the input tumor catalogue data, then the exposure intensities are estimated for each bootstrap sample.

3.3 Simulated Annealing distribution

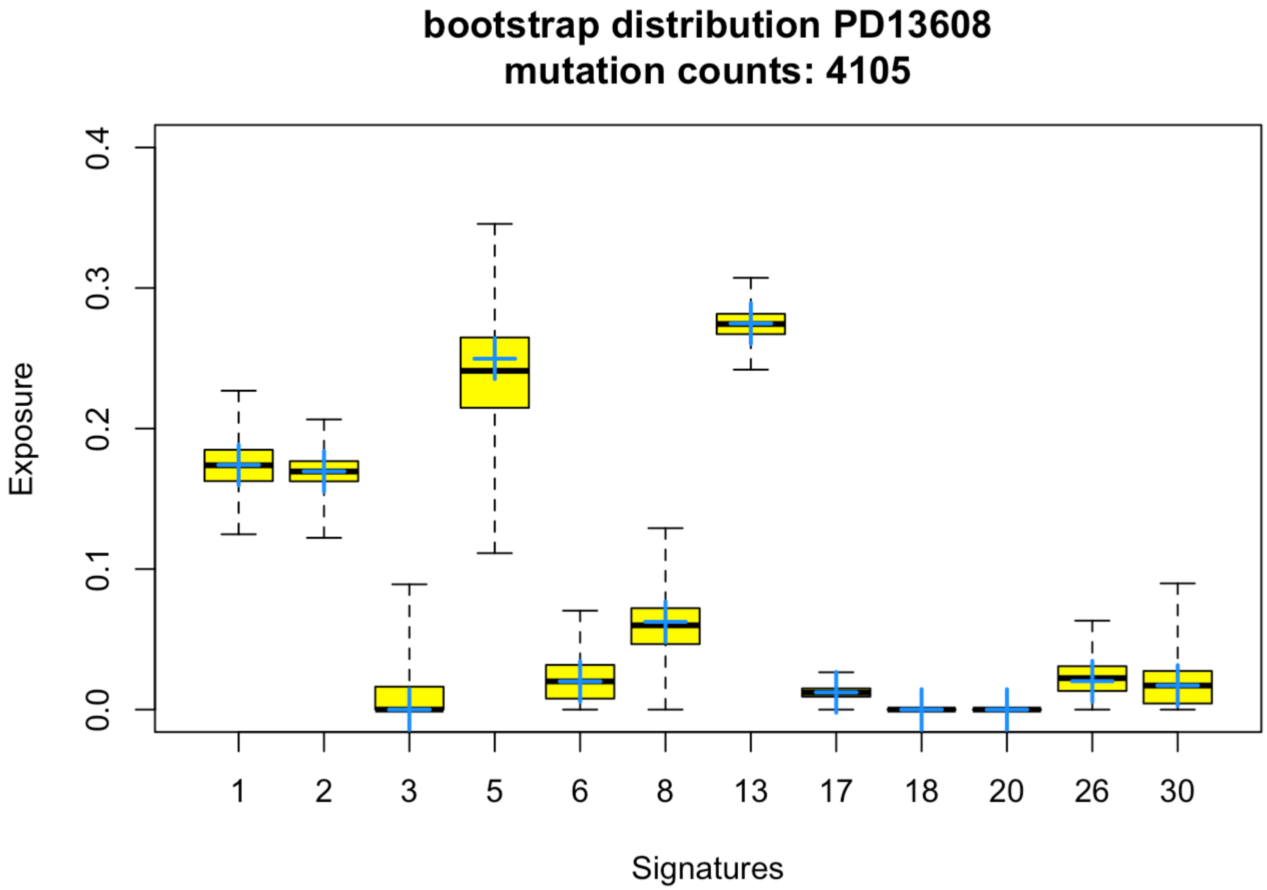
bootstrap technique is good representation of adding noise to the input data, that is, to understand how the exposure intensities will change when patients have similar tumor profiles. While biologically speaking, when we try to find optimal solutions, they may not have meaningful interpretation in biology, as the optimal solutions may be contaminated by noise. Another interesting question arises is to understand how the exposure intensities will be affected when suboptimal solutions are reached. The following figure shows using the generalized SA, with stopping threshold set at 1.01, 1.03, and 1.05 times optimal error obtained by QP respectively, to explore the global solution space. The red dots are the results from QP. The solutions of bootstrap are usually near optimal QP, and for SA, most solutions stay close to optimal. Some signatures/exposures are more variable then others, and the higher error we allow the further we diverge from optimal solution. We can also see that allowing even small error can quickly lead to overexposure or underexposure of signature, for example Signature 3 and 5 in patient PD13608, respectively, which explains why dS have high signature 3 and low signature 5 in patient PD13608 (higher errors lead to completely different exposure intensities).

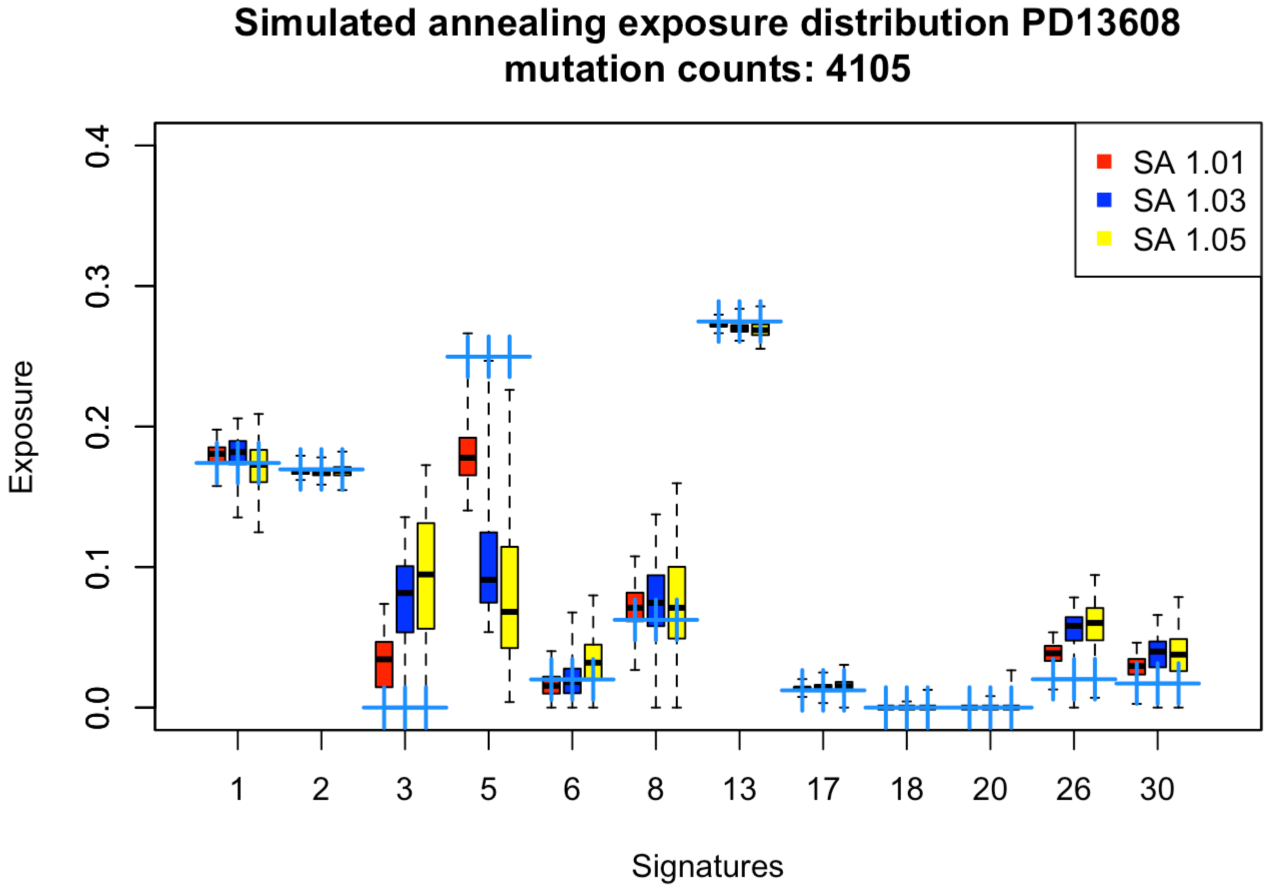
```
for (sample.i in selected_patients) {
    print(sample.i)
    # tumor catelogue
    m = tumorBRCA[, sample.i]
    ## bootstrap result
    boot = bootstrapSigExposures(m, P, R = 1000, mutation.count = size[sample.i])
    ## simulated annealing result with error 1.01*optimal error by QP
    SA101 = suboptimalSigExposures(m, P, R = 1000, suboptimal.factor = 1.01)
    ## simulated annealing result with error 1.03*optimal error by QP
    SA103 = suboptimalSigExposures(m, P, R = 1000, suboptimal.factor = 1.03)
    ## simulated annealing result with error 1.05*optimal error by QP
    SA105 = suboptimalSigExposures(m, P, R = 1000, suboptimal.factor = 1.05)


    ## boxplot for bootstrap, resampling with its own size
    boxplot(t(boot$exposures), range = 0, ylim = c(0, ymax[sample.i]), ylab = "Exposur
e",
        main = paste0("bootstrap distribution ", sample.i, "\n", "mutation counts: ",
            size[sample.i]), cex.axis = 1, xaxt = "n", xlab = "Signatures",
        col = "yellow")
    axis(1, 1:12, labels = as.character(sigsBRCA), las = 1)
    points(QP[, sample.i], pch = 3, col = "dodgerblue", cex = 2, lwd = 2)

    boxplot(t(SA101$exposures), ylim = c(0, ymax[sample.i]), range = 0, ylab = "Exposur
e",
        at = 0:11 * 4 + 1, xlim = c(0, 49), xaxt = "n", xlab = "Signatures",
        col = "red", main = paste0("Simulated annealing exposure distribution ",
            sample.i, "\n", "mutation counts: ", size[sample.i]))
    boxplot(t(SA103$exposures), range = 0, at = 0:11 * 4 + 2, xaxt = "n", add = TRUE,
        col = "blue")
    boxplot(t(SA105$exposures), range = 0, at = 0:11 * 4 + 3, xaxt = "n", add = TRUE,
        col = "yellow")
    axis(1, at = 0:11 * 4 + 2, labels = as.character(sigsBRCA), las = 1, tick = TRUE)
    legend("topright", legend = c("SA 1.01", "SA 1.03", "SA 1.05"), col = c("red",
        "blue", "yellow"), pch = 15)
    points(c(1:3, 5:7, 9:11, 13:15, 17:19, 21:23, 25:27, 29:31, 33:35, 37:39,
        41:43, 45:47), rep(QP[, sample.i], each = 3), pch = 3, col = "dodgerblue",
        cex = 2, lwd = 2)
}
```

```
## [1] "PD13608"
```


bootstrap distribution PD13608
mutation counts: 4105
Exposure
0.0
0.1
0.2
0.3
0.4
Signatures
1
2
3
5
6
8
13
17
18
20
26
30


Simulated annealing exposure distribution PD13608
mutation counts: 4105
SA 1.01
SA 1.03
SA 1.05
Exposure
0.0
0.1
0.2
0.3
0.4
Signatures
1
2
3
5
6
8
13
17
18
20
26
30

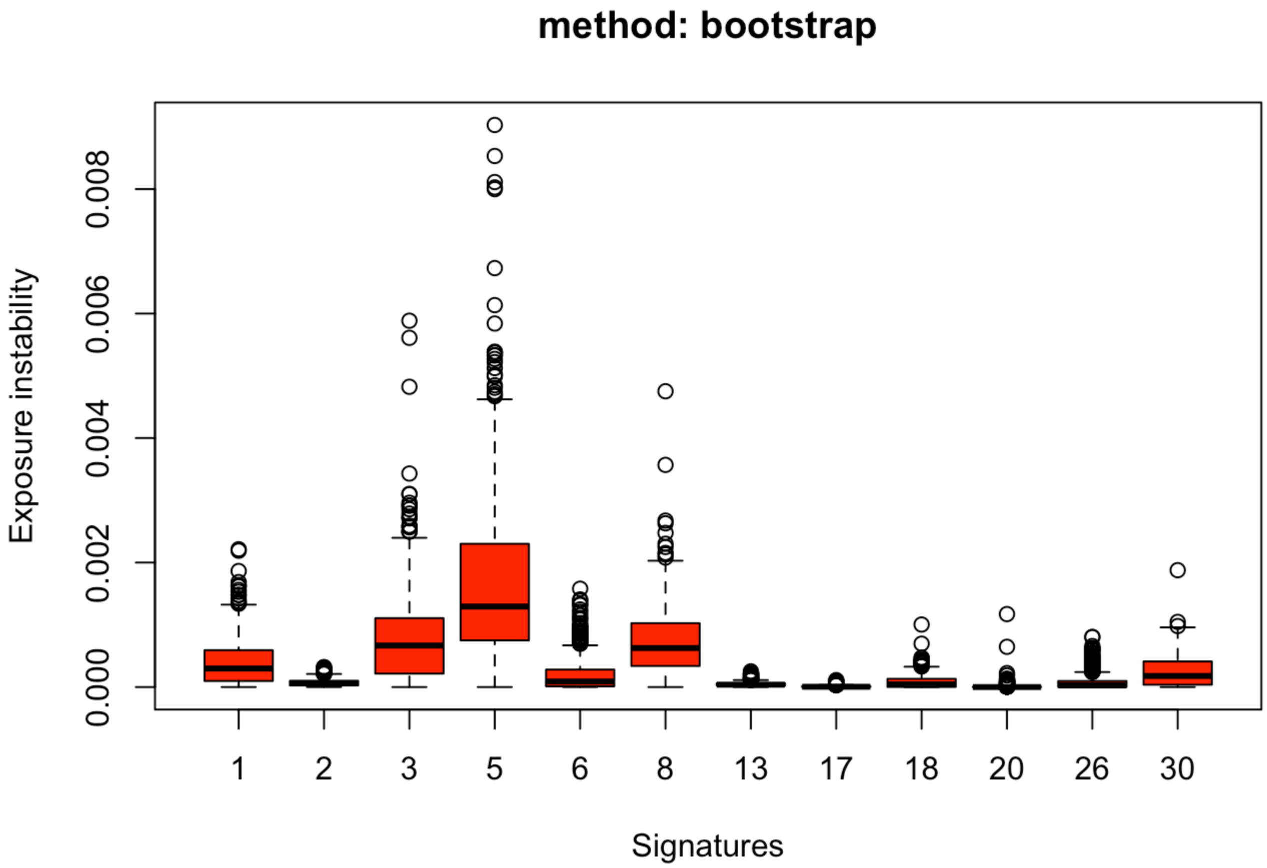
4.Extension

With the help of the confidence analysis, there are a few more things we can do. These observations open a door to assess the stability of the signatures with confidence. Additionally, it helps to detect novel signatures that have been missed in breast cancer.

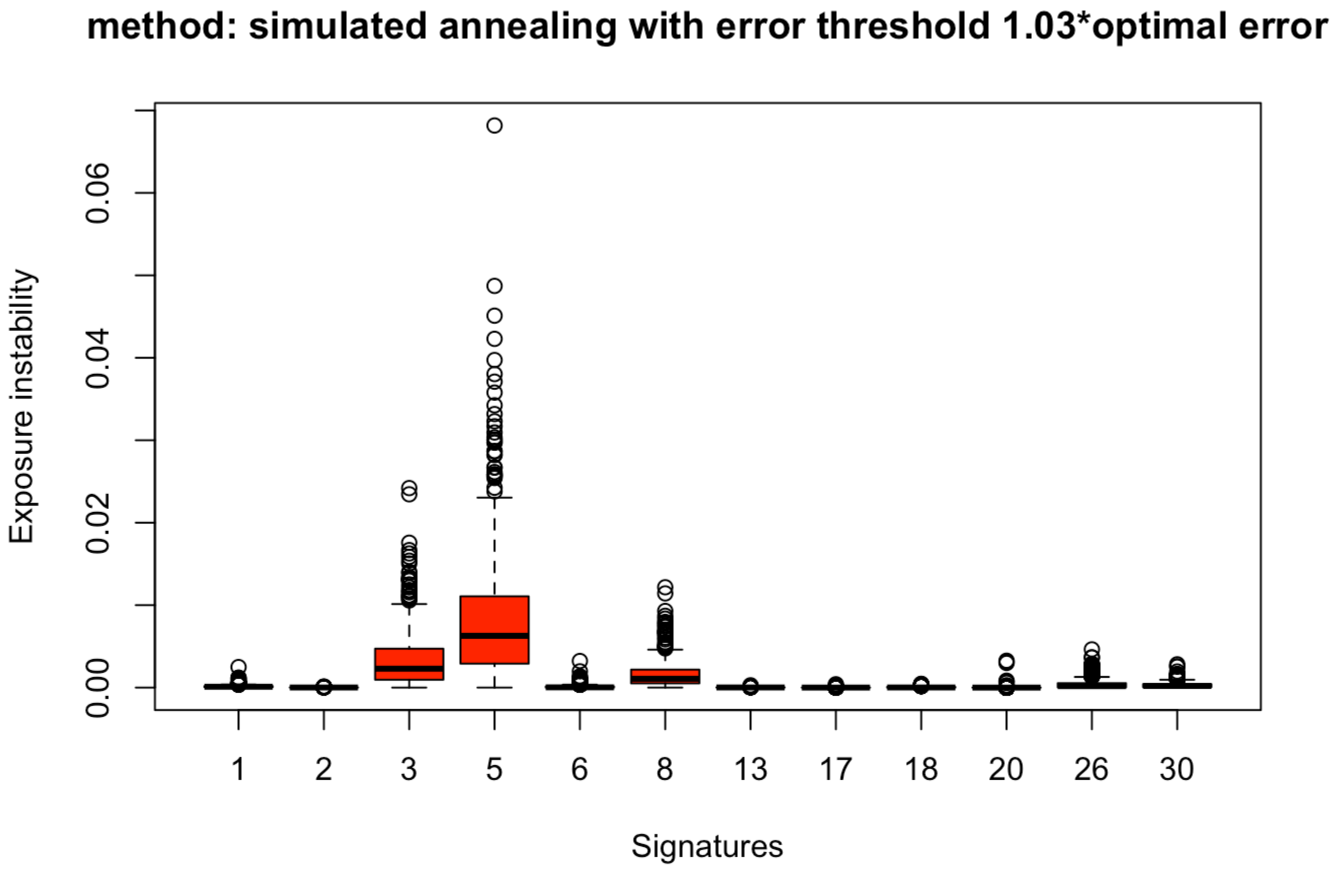
4.1 Stability analysis

We investigate the stability of the signatures in the terms of mean square error (MSE) with the mean obtained by QP, not the mean of the bootstrap or SA distribution. Using this creteria, we obtain the following plot.

```
## mse of bootstrap for all 560 patients
boot_mse = read.csv("~/Documents/ICGC/data/bootown_MSE.csv", check.names = F,
    row.names = 1)
sa103_mse = read.csv("~/Documents/ICGC/data/sa103_MSE.csv", check.names = F,
    row.names = 1)
par(mfrow = c(1, 1))
boxplot(t(boot_mse), ylab = "Exposure instability", xaxt = "n", xlab = "Signatures",
    col = "red", main = "method: bootstrap")
axis(1, 1:12, labels = as.character(sigsBRCA), las = 1)
```



```
boxplot(t(sa103_mse), ylab = "Exposure instability", xaxt = "n", xlab = "Signatures",
    col = "red", main = "method: simulated annealing with error threshold 1.03*optimal e
rror")
axis(1, 1:12, labels = as.character(sigsBRCA), las = 1)
```



4.2 detecting novel signatures

The publish work has shown that 12 signatures are present in breast cancer. With the confidence analysis here we have proposed, we decompose the tumor catalogue into all 30 signatures with p-value less than 0.01, and detect signature 9, 12 and 16 are novel signatures that prensent in breast cancer.

Then, we decompose the tumor catalogue into 15 signatures to access the genomic features and transcribed strand bias characteristics of the new detected sigantures. More details on the application of the confidence analysis can be found in our paper: Detecting presence of mutational signatures in cancer with confidence, H. Xiaoqing, D. Wojtowicz and T.M. Przytycka. (submitted).

References

Alexandrov, L. B., S. Nik-Zainal, D. C. Wedge, P. J. Campbell, and M. R. Stratton. 2013. “Deciphering Signatures of Mutational Processes Operative in Human Cancer.” Journal Article. *Cell Rep* 3 (1): 246–59. doi:10.1016/j.celrep.2012.12.008 (https://doi.org/10.1016/j.celrep.2012.12.008).

Goldfarb, D., and A. Idnani. 1983. “A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs.” Journal Article. *Mathematical Programming* 27 (1): 1–33. doi:10.1007/bf02591962 (https://doi.org/10.1007/bf02591962).

Nik-Zainal, S., H. Davies, J. Staaf, M. Ramakrishna, D. Glodzik, X. Zou, I. Martincorena, et al. 2016. “Landscape of Somatic Mutations in 560 Breast Cancer Whole-Genome Sequences.” Journal Article. *Nature* 534 (7605): 47–54. doi:10.1038/nature17676 (https://doi.org/10.1038/nature17676).

Rosenthal, R., N. McGranahan, J. Herrero, B. S. Taylor, and C. Swanton. 2016. “DeconstructSigs: Delineating Mutational Processes in Single Tumors Distinguishes Dna Repair Deficiencies and Patterns of Carcinoma Evolution.” Journal Article. *Genome Biol* 17: 31. doi:10.1186/s13059-016-0893-4 (https://doi.org/10.1186/s13059-016-0893-4).

Tsallis, C., and D. A. Stariolo. 1996. “Generalized Simulated Annealing.” Journal Article. *Physica A,* 233: 395–406.

Xiang, Y., S. Gubian, B. Suomela, and J. Hoeng. 2013. “Generalized Simulated Annealing for Efficient Global Optimization: The GenSA Package for R.” *The R Journal Volume 5/1, June 2013*. http://journal.r-project.org/ (http://journal.r-project.org/).