CNV Quality Score: framework and simulations

Kaido

6.04.2019

The framework

Consider a true copy number variant (CNV) somewhere in the genome. Let us assume its frequency in some population of interest is p. Let

$$X \sim B(1, p)$$

be a random variable modelling the presence of the CNV in individuals of the population. Usually, CNVs are called/detected using software such as PennCNV. Unfortunately, PennCNV is not always able to recover the true CNV, sometimes even calling CNVs falsely. Let $Y = f(X)$ be the random variable modelling PennCNV calls, with

$$p_{01} = \Pr(Y = 1|X = 0)$$

and

$$p_{10} = \Pr(Y = 0|X = 1)$$

denoting the false positive rate and the false negative rate of the software, respectively. Further, let $p_{00} = 1 - p_{01}$ and $p_{11} = 1 - p_{10}$ denote the true negative rate and the true positive rate. Possibly abusing the mathematical notation a little, it follows that

$$Y = f(X) = \begin{cases} 1, & \text{with probability } p_{11} \text{ if } X = 1 \\ 0, & \text{with probability } p_{10} \text{ if } X = 1 \\ 1, & \text{with probability } p_{01} \text{ if } X = 0 \\ 0, & \text{with probability } p_{00} \text{ if } X = 0 \end{cases} \sim B(1, p \cdot p_{11} + (1 - p) \cdot p_{01}).$$

We do not know the values p_{01} and p_{10} but we could estimate them. The Estonian Genome Center, University of Tartu (EGCUT) has multiple -omics data available (e.g. whole genome sequences, gene expression data, methylation data) which allowed us to validate some PennCNV calls with biological proof. The validation is probably not perfect but for the purposes of this simulation exercise, we assume it behaves like an oracle.

The average false positive rate of CNVs called by PennCNV is about 0.008 while the average false negative rate is roughly about 0.4 (**Figure 1**). It makes sense – CNVs are usually quite rare so it would be unreasonable for the software to produce too many positive calls. However, exactly because the CNVs are very rare (meaning there are a lot of individuals without the CNV), even a small false positive rate can introduce a relatively big number of false calls.

If we are interested in testing whether CNVs in some population are more or less frequent than in another population then observing the values of Y instead of X can be problematic due to the diluted/distorted signal. For example, if the frequencies of CNVs in cases and controls are $p_1 = 0.02$ and $p_0 = 0.01$, respectively, then the true odds ratio would be

$$\mathrm{OR}_X = \frac{p_1 \cdot (1 - p_0)}{p_0 \cdot (1 - p_1)} \approx 2.$$

However, if we assumed $p_{01} = 0.008$ and $p_{10} = 0.4$ then

$$\mathrm{OR}_Y = \frac{(p_1 p_{11} + (1 - p_1)p_{01}) \cdot (1 - p_0 p_{11} - (1 - p_0)p_{01})}{(p_0 p_{11} + (1 - p_0)p_{01}) \cdot (1 - p_1 p_{11} - (1 - p_1)p_{01})} \approx 1.43$$

which is a much smaller effect and thus more difficult for statistical tests to recover.

The quality score (QS) is unable to do anything with false negative calls (it would require CNV imputation) but it can reverse the false positive PennCNV calls. It works by assigning a score from $[0, 1]$ to every CNV

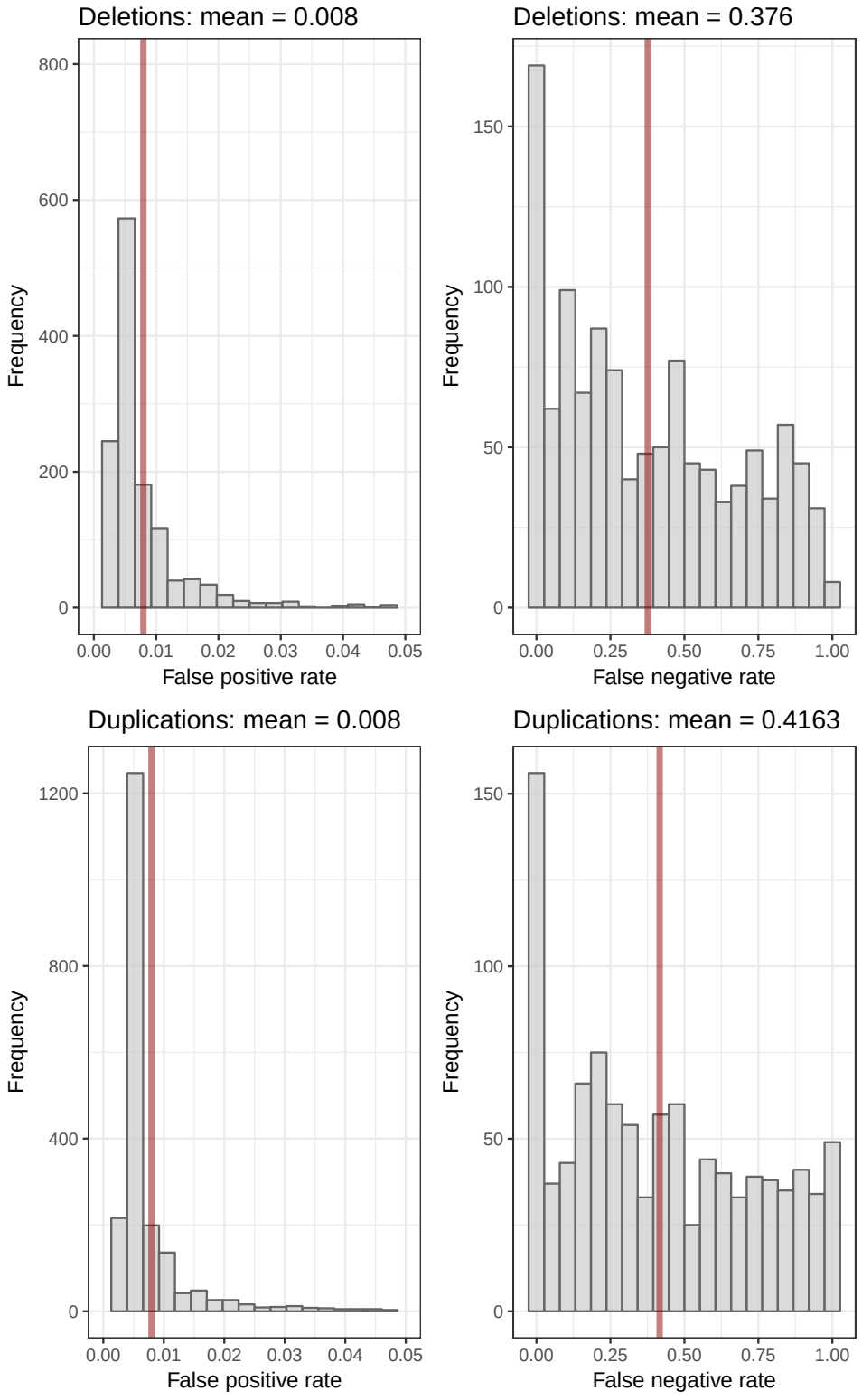

Figure 1: Average false positive rate and false negative rate of CNVs called by PennCNV in the EGCUT data. These are similar for both deletions and duplications.

called by the PennCNV software: true positive calls should get a score close to 1 while false positive scores should get a score close to 0. Let $Z = g(X, Y)$ be the random variable modelling the scores. Let $\mathcal{S}_0$ and $\mathcal{S}_1$ be the distributions from which false positive and true positive scores are generated, respectively. Possibly abusing the mathematical notation a little, we could write

$$Z = g(X,Y) = \begin{cases} S_1 \leftarrow \mathcal{S}_1, & Y = 1 \wedge X = 1 \\ S_0 \leftarrow \mathcal{S}_0, & Y = 1 \wedge X = 0 \\ 0, & Y = 0 \end{cases} = \begin{cases} S_1 \leftarrow \mathcal{S}_1, & p \cdot p_{11} \\ S_0 \leftarrow \mathcal{S}_0, & (1-p) \cdot p_{01} \\ 0, & 1 - pp_{11} - (1-p)p_{01}, \end{cases}$$

where $S_i \leftarrow \mathcal{S}_i$ means selecting a value S_i randomly from the distribution $\mathcal{S}_i$. Essentially, Z takes a random value from $\mathcal{S}_1$ with probability $p \cdot p_{11}$, a random value from $\mathcal{S}_0$ with probability $(1-p) \cdot p_{01}$ and is 0 in case of every negative PennCNV call.

We do not know the distributions $\mathcal{S}_1$ and $\mathcal{S}_0$ but again we can estimate these based on the EGCUT data. Histograms with bin width 0.02 in **Figure 2** represent exactly these distributions empirically. We can approximate these with very simple step functions with two steps, resulting in the probability density functions

$$f_{S_1}(x) = \begin{cases} 25.5, & 0 < x \leq 0.02 \\ 0.5, & 0.02 < x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

and

$$f_{S_0}(x) = \begin{cases} 10.8, & 0.98 < x \leq 1 \\ 0.8, & 0 < x \leq 0.98 \\ 0, & \text{otherwise} \end{cases}$$

for $S_1 \leftarrow \mathcal{S}_1$ and $S_0 \leftarrow \mathcal{S}_0$, respectively.

It is easy to generate random values from these distributions. To get values from $\mathcal{S}_1$, we need to generate from $\mathcal{U}(0, 0.02)$ with probability $0.02 \cdot 25.5$ and from $\mathcal{U}(0.02, 1)$ with probability $(1 - 0.02) \cdot 0.5$. To get values from $\mathcal{S}_0$, we need to generate from $\mathcal{U}(0.98, 1)$ with probability $(1 - 0.98) \cdot 10.8$ and from $\mathcal{U}(0, 0.98)$ with probability $0.98 \cdot 0.8$.

Of course, we could try to find a better fit for $\mathcal{S}_1$ and $\mathcal{S}_0$, e.g. by increasing the number of steps. However, the proposed fit has some positive properties. First, it is very simple and thus more likely to be generalizable to cohorts other than EGCUT. Second, the interpretation is straightforward: we can detect false positive calls roughly 50% of the time and true positive calls roughly 20% of the time, all other times we generate a random score uniformly over the unit interval. Given the above, it is actually of great interest whether the QS increases power to detect variable CNV effects in cases versus controls under the alternative hypotheses.

Simulations

We can actually find an exact result for the discrete binomial distributions. Let $X_{1,i} \sim B(1, p_1), i = 1, ..., n_1$ and $X_{0,i} \sim B(1, p_0), i = 1, ..., n_0$ be two IID samples from specified Bernoulli distributions. Then

$$T_1 = \sum_{i=1}^{n_1} X_{1,i} \sim B(n_1, p_1)$$

and

$$T_0 = \sum_{i=1}^{n_0} X_{0,i} \sim B(n_0, p_0),$$

respectively. Controlling for type 1 error α,

$$\text{Power} = \Pr(\mathrm{P}_t < \alpha | p_1 \neq p_0) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_0} \Pr(T_1 = i, T_0 = j) \cdot \mathbb{1}_{\mathrm{P}_{t,i,j} < \alpha},$$

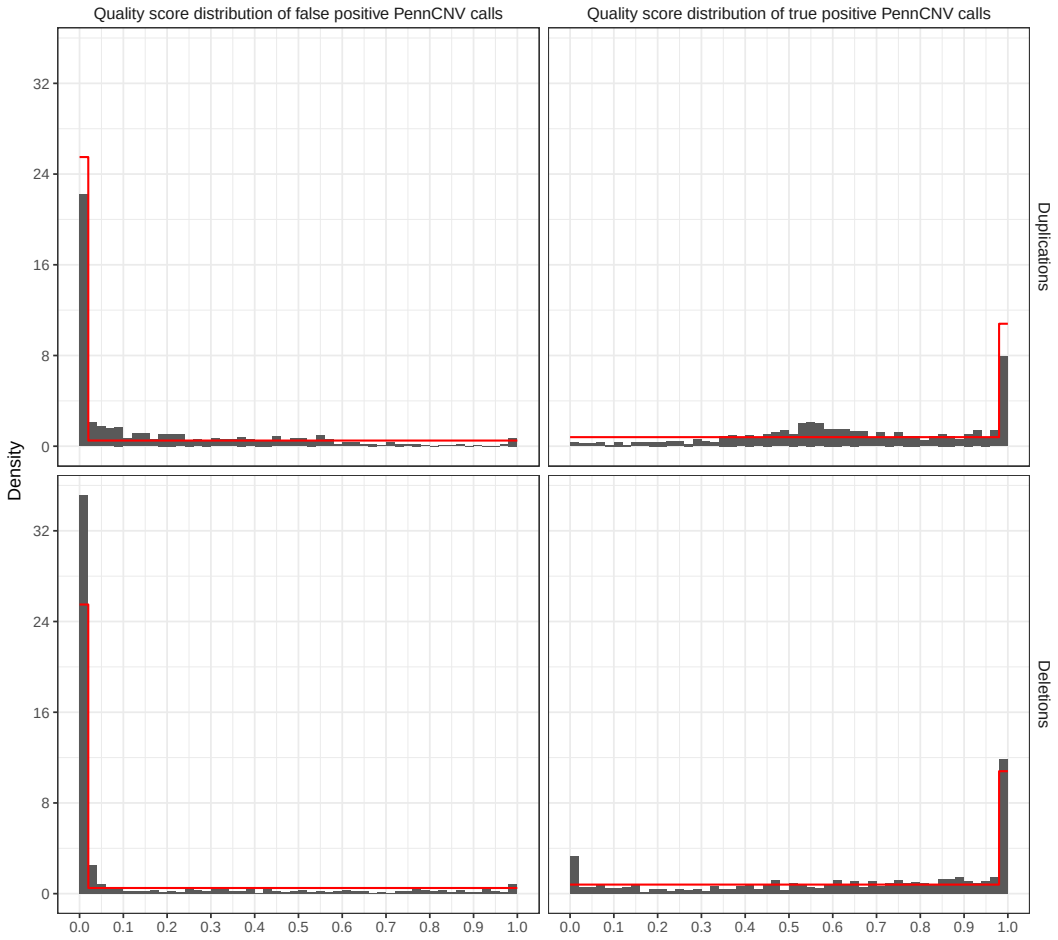

Figure 2: Histograms with bin width 0.02 representing distributions $\mathcal{S}_0$ (left) and $\mathcal{S}_1$ (right). These are roughly the same for both duplications and deletions. The red line indicates a simple approximation by step functions of two steps (defined to be the same for both duplications and deletions).

where P_t is a random variable following a p-value distribution of statistical test t under the above configuration, $\mathrm{P}_{t,i,j}$ is the p-value produced by the statistical test under $T_1 = i$ and $T_0 = j$, and $\mathbb{1}$ denotes an indicator function. It is reasonable to expect that CNV carriers in one population are independent from the carriers in the other population, therefore

$$\Pr(T_1 = i, T_0 = j) = \Pr(T_1 = i) \cdot \Pr(T_0 = j) = \binom{n_1}{i} p_1^i (1-p_1)^{n_1 - i} \cdot \binom{n_0}{j} p_0^j (1-p_0)^{n_0 - j},$$

which we can easily calculate for every pair (i, j), $i = 1, ..., n_1$, $j = 1, ..., n_0$. We can then use the statistical test to evaluate whether the null hypothesis $p_1 = p_0$ holds for every pair with probability mass greater than 0.

To estimate power using the QS, we need to take many times (m) random samples $Z_{1,i}, i = 1, ..., n_1$ (assuming true CNV frequency p_1, false negative rate p_{10}, and false positive rate p_{01}) and $Z_{0,i}, i = 1, ..., n_0$ (assuming true CNV frequency p_0, false negative rate p_{10}, and false positive rate p_{01}). Then

$$\text{Power} = \Pr(\mathrm{P}_t < \alpha | p_1 \neq p_0) \approx \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\mathrm{P}_{t,i} < \alpha},$$

where P_t is a random variable following a p-value distribution of statistical test t under the above configuration (Z_1 and Z_0 with parameters p_1, p_0, p_{01}, p_{10}), and $\mathrm{P}_{t,i}$ is the p-value produced by the statistical test in the i-th simulation run.

Simulation results

The quality score indeed seems to increase power over the raw PennCNV binary variable by roughly 5-10% if the CNV frequency is low (**Figure 3**). The same benefit does not seem to be there if the CNV is more frequent (**Figure 4**). Note that the total sample size was fixed at $N = 33000$.

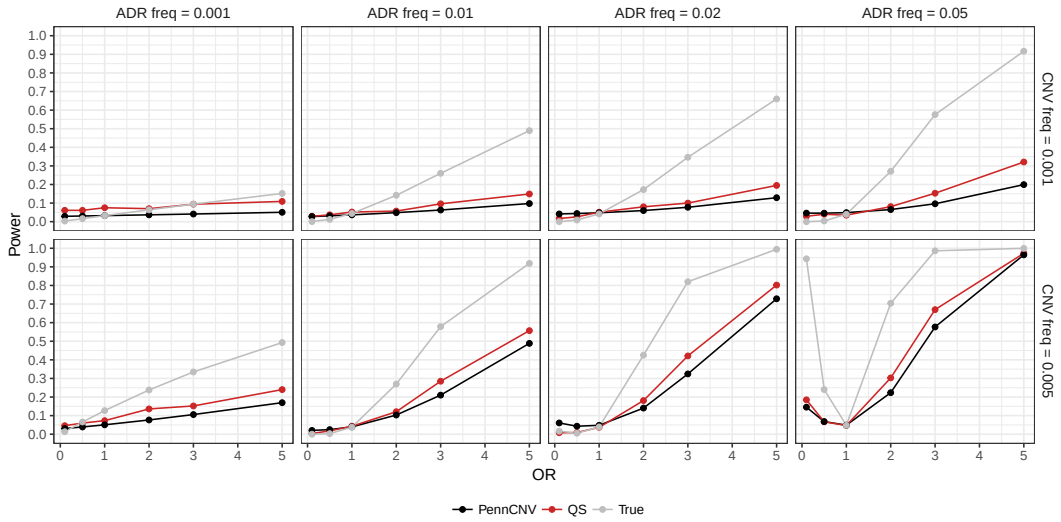

Figure 3: Power curves in case of low CNV frequencies and variable disease (ADR is actually an abbreviation for adverse drug reaction) frequencies. The gray line corresponds to using the true CNV random variable X, the black line corresponds to using Y and the red line corresponds to using the quality score Z. The total sample size was fixed at $N = 33000$.

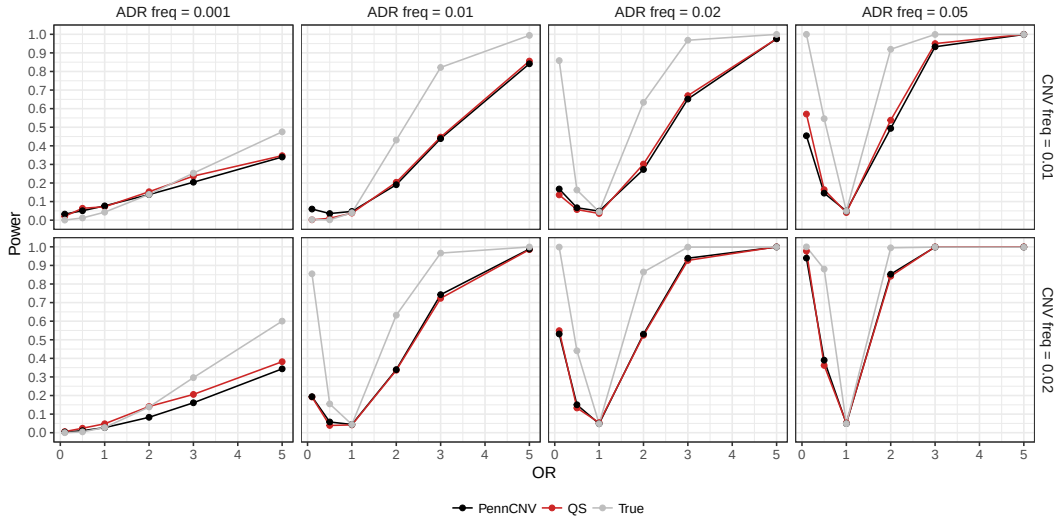

Figure 4: The same as **Figure 3** but in case of higher CNV frequencies.