COMPUTATIONAL METHODS

Latin Square Design

Our Latin Square Design (LSD) for experimental planning of multiplexed studies is based on the principle of Latin squares[51] and should more formally be called a *binary semi-magic square*. In order to fully understand the principles of this design, we first introduce some formal notations. We consider an experimental planning as a $f \times r$ matrix $\boldsymbol{E}$ of binary integers, with $f \in \mathbb{R}$ being the number of available forward tagged primers and $r \in \mathbb{R}$ the number of reverse tagged primers. To simplify further computations, we also only consider square design matrix, i.e. $f = r$. Each entry of this matrix $\boldsymbol{E} = \left\{e_{ij}, i \in \{1 \dots f\}, j \in \{1 \dots r\}\right\}$ is a binary integer that indicates if the corresponding tagged primer combination is used to amplify a sample ($e_{ij} = 1$) or not ($e_{ij} = 0$). We introduce here two terms that will be used for defining the constraints. First, the *saturation* of the experimental design can be computed as the ratio of samples multiplexed over the number of tagged primer combinations, i.e. $saturation = {\sum_i \sum_j e_{ij}}/{f \times r}$. The saturation is an indicator of the usage density of a design (and its sensitivity to mistagging) as $saturation \in]0,1]$ with a saturation of 1 being a fully saturated design (all tagged primer combinations are used). Second, the primer usage indicates for each tagged primer, the number of samples for which this particular tagged primer is used, i.e. $\mathcal{P}_i = \sum_k e_{ik}$ (here for a forward primer). The goal of the LSD is to reduce the saturation while balancing the primer usage frequency. Hence, finding a solution can be seen as an integer optimization problem with the objective function being that we want to find a tagged primer combination for all of our samples

$$\sum_{i=1}^{f} \sum_{j=1}^{r} e_{ij} = \mathcal{N}_{samples}$$

with the main constraints being that we want the sample to be equally distributed over all tagged primers, ie. each primer is used the same number of times

$$\sum_k e_{ik} = \sum_k e_{kj} = \mathcal{P}_\alpha, \forall i, j$$

As we are fully aware, this problem can only be solved if the primer usage $\mathcal{P}_\alpha$ is an integer divisor of the number of samples $\mathcal{N}_{samples}$, i.e. $\mathcal{P}_\alpha \equiv 0\left[\mathcal{N}_{samples}\right]$. In order to solve the particular cases where the number of samples is not an integer divisor, we simply take the design of superior saturation (being the next ceiling integer $\lceil \mathcal{P}_\alpha \rceil$ of the primer usage frequency). Then we iteratively remove samples in the regions of the design with the higher local saturation until we obtain the required number of samples.

Even with all these constraints, we can find a whole set of different matrix (and, therefore, designs) that fits these requirements. Therefore, in order to provide the best problem-dependent experimental design, we select the design matrix based on a set of constraints that depend both on the nucleotide sequence of the tag primers and the samples that are being sequenced. If we denote as $\mathcal{L}$ the set of Latin Square Designs that fit the previous requirements, we select the design $l_{best} \in \mathcal{L}$ that minimize the proximity of tags with biggest sequence similarity, i.e.

$$l_{best} = \underbrace{argmin}_{\mathcal{L}} \left\{ \sum_{ij} \left(\mathcal{D}_{NW}\left(primer_i^{seq}, primer_J^{seq}\right) \times e_{ik} \times e_{ik} \right) \right\}$$

with $\mathcal{D}_{NW}$ being the Needleman-Wunsch distance between tagged primers i and j. As we can see, the distance is accounted only if the two forward primers are used with a common reverse primer k. Finally, the samples can be distributed over the design by favouring that similar samples (in terms of composition) should be amplified preferably with a shared primer.

ISU-Based filtering

The ISU-Based filter is designed to discriminate biases (mostly stemming from non-random errors) from true sequences. The idea of this filter is to shift the focus of filtering process towards an ISU-centred approach.

Following the notations of the previous section (LSD), we still consider the matrix of multiplexing with each entry $e_{ij}, i \in \{1 \dots f\}, j \in \{1 \dots r\}$ displaying the use of a particular tagged primer combination in the experimental design. We also denote the set of sequences found with the tagged primers i and j as seq_{ij}. Following these notation, sequences are non-critical mistags if $ISU_k \in seq_{ij}$ with $e_{ij} = 0$. We denote here $\left\| ISU_k^{ij} \right\|$ as being the read abundance of ISU_k in sample (i,j). The idea behind the ISU-based filtering is to rely on the abundance of each sequence found as non-critical mistags as a baseline of abundance for the mistagging. Our rationale is that if a sequence is found in a true sample with a level of abundance identical to those in non-critical mistags, then there is a high probability of this sequence to be a critical mistag. Therefore, we gather for each sequence the distribution of non-critical abundance in unused samples that share at least one primer with the current sample. More formally if we are trying to assess if a sequence ISU_k^{ij} is a bias in sample (i,j), we compute the set

$$\mathcal{D}_{nc} = \left\{ \left\| ISU_k^{lm} \right\| \middle| (e_{lm} = 0) \wedge (l = i \vee m = j) \right\}$$

It should be noted that we collect only the abundance of non-critical mistags (without accounting for the same ISU in other true samples). Then, we need to define if the abundance of this ISU inside the

current (true) sample is within the same range as all the non-critical mistags. This would indicate that the current ISU is only a part of the overall level of "noise" and, therefore, that this ISU is actually a critical mistag. However, if the abundance of the ISU in the true sample clearly deviates from the distribution of non-critical mistags, then it clearly points out to the current ISU being a true part of the current sample (and a fortiori that this ISU generated the corresponding non-critical mistags). In order to perform this discrimination, we rely on the Modified Thompson τ −Test. Hence, we compute the mean $\mu_{\mathcal{D}_f}$ and standard deviation $\sigma_{\mathcal{D}_f}$ of the distribution $\mathcal{D}_f = \{\mathcal{D}_{nc} \cup \left\| ISU_k^{ij} \right\|\}$ (distribution including the abundance of non-critical mistags and abundance of the current ISU). Then, we compute the deviation of the current ISU abundance to the mean abundance of the distribution $\delta_k = \left| \left\| ISU_k^{ij} \right\| - \mu_{\mathcal{D}_f} \right|$. Finally, we compute a rejection region based on the modified Thompson τ value

$$\mathcal{R} = \tau . \sigma_{\mathcal{D}_f} = \frac{t_{\alpha/2} . (n-1)}{\sqrt{n}\sqrt{n-1+t_{\alpha/2}^2}} . \sigma_{\mathcal{D}_f}$$

with $t_{\alpha/2}$ being the critical student's t test value and $n = \left\| \mathcal{D}_f \right\|$, the number of elements in the distribution. If the deviation of abundance of the current ISU is outside of the rejection region ($\delta_k > \mathcal{R}$), then it is an outlier and, therefore, is considered as a correct ISU. If the abundance is inside the rejection region ($\delta_k \leq \mathcal{R}$), the ISU is discarded. It is very interesting to note, that we actually reverse the use of such tests. In our case, an outlier is a good sign (as it shows that the ISU "stands out" from the overall noise). The advantage of this test in our context is manifold. First, it is a parameter-free method, which allows bypassing the use of any subjective parameter. Moreover, it accounts for the distribution average as well as its standard deviation, which provides a statistically determined criterion for filtering sequences. The global workflow showing the algorithmic steps is displayed

```
1.    foreach read in rawFastQ
2.       % Filter the reads based on mean quality and ambiguous bases
3.       filterLowQualitySequences()
4.       filterAmbiguousBases()
5.       % Try to find the tagged primers in the read
6.       (curForward,curReverse) = findPrimersByAlignment
7.       if (curForward,curReverse) is in multiplexedSamples
8.          % Current sequence is part of a real sample
9.          putReadInRealSample(read, sample)
10.      else
11.         % Current sequence is a non-critical mistag
12.         putReadInNonCriticalSample(read, sample)
13.   % At the end, compute the abundance of each ISU in each sample
14.   dereplicateSetAndComputeAbundance(samples)
15.   % Now we process each of the real samples
16.   foreach sample in multiplexedSamples
17.      (curForward, curReverse) = tagCombination(sample)
18.      % We perform a decision for each ISU in the sample
19.      foreach ISU in sample
20.         % Find the abundances of current ISU tagged with the current forward
            tag and any reverse tag
21.         nonCriticalForward = findAbundances(ISU, curForward)
22.         % Same with the reverse tag (and any forward tag)
23.         nonCriticalForward = findAbundances(ISU, curReverse)
24.         allNonCriticalAbundances = [nonCriticalForward,nonCriticalReverse]
```

```
25.         % Perform the modified Thompson tau-test
26.         curAbundance = abundance(ISU, curForward, curReverse)
27.         totalAbundance = [curAbundance, allNonCriticalAbundances]
28.         % Properties of the distribution (mean, deviation, size)
29.         meanDistrib = mean(totalAbundance)
30.         devDistrib = std(totalAbundanceDistribution)
31.         n = length(totalAbundanceDistribution)
32.         % Deviation of current abundance to mean distribution
33.         curDeviation = abs(meanDistrib - curAbundance)
34.         % Rejection region
35.         r = ((tAlph*(n-1))/(sqrt(n)*sqrt(n-1+(tAlph^2))))*devDistrib;
36.         if (curDeviation <= reject)
37.            % Current sequence is a critical mistag
38.            dismissSequence
39.         else
40.            % Current sequence is not a mistag
41.            keepInFinalSet(ISU, sample)
```