Scholarly Paper

Optimized Probe Selection for Pan-genomic DNA Microarrays

Adam M. Phillippy[1*]

[1]Center for Bioinformatics and Computational Biology, Department of Computer Science, University of Maryland, College Park, MD 20742, USA

ABSTRACT

Motivation: Array comparative genomic hybridization is a quick and cheap method for detecting and genotyping unknown microbial isolates. However, there are a fixed number of probes per array, and therefore the number of loci that can be targeted by a single array is limited. For accurate strain genotyping, an array must query a fully representative set of genes from the species' pan-genome. Prior genotyping arrays have only targeted a single strain or the conserved sequences of gene families.

Results: This paper presents a new probe selection algorithm (PanArray) that can target multiple whole genomes in a minimal number of probes. Unlike arrays built on clustered gene families, PanArray guarantees that every subsequence of the genomes is independently targeted by a full complement of probes, increasing the flexibility and accuracy of the associated comparative analysis and genotyping. The viability of the algorithm is demonstrated by the design of a 385,000 probe array that fully tiles the genomes of 20 different *Listeria monocytogenes* strains at greater than two-fold coverage.

Availability and Implementation: The PanArray design software is implemented in C++, and the PanArray source code and the *L. monocytogenes* array design are freely available upon request.

Contact: amp@umiacs.umd.edu

1 INTRODUCTION

As one of their many diverse roles, DNA microarrays can be used to characterize both large-scale and small-scale genetic variation. In human cancer studies, array comparative genomic hybridization (aCGH) is commonly used to genotype cell lines and detect gene loss and copy number variations (Pinkel, et al., 1998). At a finer resolution, microarrays are also used to detect single nucleotide polymorphisms at targeted loci (Wang, et al., 1998). In addition to human screens, microarrays have been widely used for the detection and genotyping of microbial species. Notably, a viral genotyping microarray (Wang, et al., 2002) was one of the methods used to etiologically link severe acute respiratory syndrome (SARS) to a novel coronavirus (Ksiazek, et al., 2003). Arrays for the detection and comparative analysis of bacterial genomes have also been developed, including arrays for *Listeria monocytogenes* (Borucki, et al., 2004; Call, et al., 2003; Doumith, et al., 2004; Volokhov, et al., 2002; Zhang, et al., 2003). However, these earlier, low density arrays did not contain enough probes to target the entire genome of the bacterium, and were forced to probe only a small subset of the known genes.

As the density of DNA microarrays increased, it became possible to probe the entire genome of an organism in addition to only specific genes. Such an array is commonly referred to as a *whole-genome tiling array* (Mockler, et al., 2005). In the human genome, tiling arrays are designed to probe the genome at evenly spaced intervals. This creates an optimization problem in choosing which sequences should be included on the array (Bertone, et al., 2006; Graf, et al., 2007). To maximize the specificity of the array, repetitive probes should be avoided and experimental conditions, such as melting temperature, equalized. In smaller, microbial genomes, these concerns can be largely ignored, because it is possible to probe every position of the genome without leaving any gaps. For instance, Roche NimbleGen can presently synthesize 2.1 million variable length probes on a single chip. For an average 2 Mbp sized bacterium and 50 nt probe length, this would be an equivalent redundancy of about 50x—meaning that every base-pair of the genome could be spanned by 50 individual probes.

Tiling arrays have traditionally been constructed from a single reference strain and used to locate differences contained in the experimental strains. However, they can only detect and analyze sequences similar to those included on the array, and cannot discover genes unique to the experimental strains. After the introduction of the pan-genome concept (Medini, et al., 2005; Tettelin, et al., 2005), it has become increasingly clear that a single microbial species can contain a vast genomic diversity, and it is not suitable to compare against only a single reference strain. The pan-genome hypothesis states that any given species has two sets of genes. First, a set of *core* genes present in all strains that define the species; and second, a set of *dispensable* genes present in only one or a few of the strains that presumably mediate adaptation. A single genome describes the genetic material for a particular strain, but the pan-genome describes the genetic makeup for an entire species. Single reference tiling arrays cannot survey this full diversity. Ideally, a genotyping array would test for the genetic material of the entire pan-genome and not just one particular strain.

With the explosion in microarray densities, it is now possible to design pan-genome tiling arrays that contain all genetic sequence from the known pan-genome. The simplest strategy is to fully tile the genomes of each strain. However, due to similarities between the strains, some sequences would be tiled with excessive redundancy, and this approach would be very cost ineffective. Instead, a pan-genome array should aim to minimize costs by using the minimal probe set necessary to target every gene in the pan-genome with adequate coverage. The typical approach is to group

*To whom correspondence should be addressed.

individual genes into gene families and then target only the conserved sequences of those families (Chung, et al., 2005; Feng and Tillier, 2007; Willenbrock, et al., 2007). For example, Willenbrock et al. designed a 32 strain *Escherichia coli* pan-genome array by clustering homologous genes based on pairwise alignment similarity (Willenbrock, et al., 2007). Homology was defined as gene alignments with an E-value < 10^{-5}, a bitscore > 55, and alignment coverage of at least 50% of the gene length. For each resulting gene group, a consensus sequence was generated via multiple alignment, and probes were designed to target the most conserved regions of the consensus. The resulting array comprised 224,805 probes, targeting 9,252 gene groups, with a median coverage of 27 probes per gene group.

Ideally, a pan-genome array would have a probe set targeting each functionally distinct gene family in the pan-genome. However, defining functional similarity based on alignment similarity has some disadvantages. First, arbitrary thresholds for homology must be chosen. It is unclear that the chosen sequence similarity thresholds will correspond with functional similarity. Secondly, and even if a justifiable homology threshold is available, it is unclear how to properly group the genes. Clustering the genes into homogeneous groups requires some knowledge of their functional role, but such information is often unavailable. Therefore, the "align and combine" method could inadvertently group genes with different functions into a single group, thereby limiting the analysis power of the array. Finally, because probes are only designed to target annotated genes, unannotated genes and transcripts will remain undiscovered, and any future update of the annotation will outdate the array.

To address the limitations of prior pan-genome array designs, this paper describes an alternative approach that both minimizes the cost of the array and guarantees that all genes in the pan-genome are targeted by the array. The traditional gene-centric homology clustering is abandoned in favor of a more concrete, probe-centric approach. In addition, to circumvent annotation deficiencies, a whole-pan-genome tiling is designed to target all sequences, and not just the currently annotated genes. To summarize the new approach, let P be the non-redundant set of all length k subsequences from the entire pan-genome sequence G. For some probe p, let p' be its reverse complement. A candidate probe $p \in P$ is said to *cover* a location $g \in G$ if some subsequence of G containing g would effectively hybridize with p' on the array. The *Pan-Tiling* problem is to find the smallest subset $H \subseteq P$ such that every location in the pan-genome is covered by at least x probes in H (Figure 1).

The probe-centric formulation of the *Pan-Tiling* problem has two primary advantages. First, because it is unbiased with regard to genes, it overcomes deficiencies in the annotation and can be applied to draft genomes lacking annotation. New annotations can simply be remapped to the probes, without having to redesign the array. Secondly, covering every location of the pan-genome bypasses the problem of defining homologous gene sets—every gene from every strain is fully tiled with probes. Constructing a full tiling of the pan-genome seems like it would require a large number of probes, but by leveraging the similarities between strains and the non-specific nature of hybridization, a probe set can be constructed that fully covers a large pan-genome with adequate redundancy. The key to this strategy is choosing probes that will hybridize to as many of the strains in the species as possible, while using unique probes only when necessary to tile strain-specific sequences.

G =
AAAAAACCCCCCGGGGGGTTTTTT
AAAAAACCC**G**CCGGGGGGTTTT**AA**
AAAAAACCCCCCGGGGGGTTTT**AA**

P =
AAA,AAC,ACC,CCC,CCG,CGG
GGG,GGT,GTT,TTT,C**G**C,**G**CC
TT**A**,T**AA**

H =
AAA,CCC,GGG,TTT,**G**CC,T**AA**

Fig. 1. Example pan-genome G made up of three miniature genomes with differences shown in bold. Set of all 3-mer subsequences is given by P. H is the minimum subset of P that tiles G at 1x coverage. If G is double stranded and reverse complement targets are acceptable, H = {AAA,CCC,GCC,TAA}.

The methods presented in this paper were developed to aid the design of a pan-genome CGH tiling array for *Listeria monocytogenes*. *L. monocytogenes* is the causative agent of listeriosis and is a NIAID category B biodefense agent. It is particularly troublesome to the food industry because it is widely present in plant, soil, and water samples, and can grow at chill temperatures in common foods such as meats, dairy products, and seafood (Farber and Peterkin, 1991). *L. monocytogenes* is particularly well suited for pan-genome array design because there are a remarkable number of strains that have been sequenced. At the time of chip design, a total of 20 *L. monocytogenes* genome sequences were available (Table 1). The species can be divided into three primary lineages, with the sequencing effort targeting mostly lineage I and lineage II strains. The sequence conservation between the sequenced strains is not exceptional, and ranges between 94% and 99% nucleotide identity versus the reference EGD-e strain. The pan-genome CGH array described in the Results section was successfully constructed from this diverse gamut of strains.

2 METHODS

The general strategy of the PanArray design algorithm is best summarized by an analogy to the well-known *Set Cover* and *Hitting Set* problems in computer science. Let P be a set of n points and $F = \{P_1, P_2, \ldots, P_m\}$ be a family of m subsets of P. *Hitting Set* is the problem of selecting the minimum subset $H \subseteq P$ such that every set in F contains at least one element of H. *Set Cover* is the well-known dual of this problem. Although *Hitting Set* and *Set Cover* are known to be NP hard problems, a polynomial-time greedy algorithm is known to give essentially the best possible approximation (Feige, 1998; Johnson, 1973).

To see the similarities between the *Pan-Tiling* and *Hitting Set* problems, let the sequence G be a concatenation of all the genomes from a pan-genome, and let $S = \{s_1, s_2, \ldots, s_m\}$ be the set of m intervals that results from segmenting G into equal, length l segments. Let P be the non-redundant set of length k subsequences from G. A probe candidate $p \in P$ is said to *hit* a segment $s \in S$ if a match between p and a subsequence of G begins in the interval s. Let $P_i \subseteq P$ be the subset of probes that hit the segment s_i, and $F = \{P_1, P_2, \ldots, P_m\}$ for the m segments of S. The hitting set H is the smallest

Table 1. Genomic sequences included on the *Listeria monocytogenes* pan-genome tiling array. Sequences were obtained from GenBank and annotations from NMPDR and JCVI CMR. The final column shows the nucleotide identity of a whole-genome alignment versus strain EGD-e.

Strain	Lineage	Serotype	No. Bases	No. Contigs	No. Genes	EGD-e Identity
EGD-e	II	1/2a	2,944,528	1	3,002	100.00
LO28	II	1/2c	2,910,810	529	5,078	99.30
FSL F2-515	II	1/2a	2,586,267	1,415	NA	98.41
FSL J2-003	II	1/2a	2,878,206	406	4,686	98.22
1/2a F6854	II	1/2a	2,950,285	133	3,028	98.01
FSL N3-165	II	1/2a	2,886,689	33	2,963	97.52
J2818	II	1/2a	2,971,223	38	3,270	97.19
F6900	II	1/2a	2,958,319	35	3,333	97.15
J0161	II	1/2a	3,051,828	51	3,252	97.09
10403S	II	1/2a	2,866,709	32	2,944	96.90
FSL J2-064	I	1/2b	2,899,431	327	3,914	94.69
4b H7858	I	4b	2,972,254	181	3,187	94.54
FSL J1-175	I	1/2b	2,902,346	357	4,559	94.49
FSL N1-017	I	4b	2,857,865	77	3,465	94.30
HPB2262	I	4b	3,006,068	75	3,319	94.01
FSL J1-194	I	1/2b	2,986,227	44	3,792	93.98
4b F2365	I	4b	2,905,187	1	2,987	93.87
FSL R2-503	I	1/2b	3,001,696	54	4,863	93.73
FSL J2-071	IIIA	4c	3,149,923	46	3,789	93.28
FSL J1-208	IIIB	4a	2,260,760	1,494	NA	92.84

possible subset $H \subseteq P$ such that each P_i shares at least one element with H. Therefore, the solution to this *Hitting Set* problem is a minimum subset of probes H such that every segment of the pan-genome is hit by at least one probe in H. Therefore, the probes in H effectively tile the entire segmented pan-genome using a small number of probes. This *Hitting Set* formulation is not equivalent to the original *Pan-Tiling* problem definition, but the imposition of fixed segments on the pan-genome is a helpful simplification.

2.1 Probe Indexing

Segmenting the pan-genome sequence may prohibit finding an optimal solution to the *Pan-Tiling* problem, but it does not limit adjustments to the coverage of the tiling. For a segment length of l, segments are simply marked off every l bases of the pan-genome—with the first segment s_1 covering the interval $[1, l]$, and the second segment s_2 covering $[l+1, 2l]$, and so on. Segments extending across contig boundaries are discarded. For a segment length l equal to the probe length k, the resulting depth of coverage averages 1-fold because the probes are spaced k bases apart on average. To adjust the resulting coverage, the fixed segment size can be modified beforehand and the resulting depth of coverage c is expected to be $c \approx k / l$. The extreme case being $l = 1$, which results in exactly k-fold coverage because a probe must be selected for every position in G.

Once the pan-genome is discretized into a set of segments, each segment must be mapped to the set of probes it contains. As before, a probe p hits a segment if a match between p and G begins within the segment's interval. A match can be defined by any criteria necessary for efficient hybridization. To help reduce probe redundancy, PanArray defines a match as a full-length alignment with either 0 or 1 mismatches. Any suitable k-mer indexing algorithm can be utilized for this phase, but allowing for mismatches can be computationally expensive. The PanArray software uses a fast, but memory intensive, compressed keyword tree for indexing all probe hits. Alternatively, a slower, but memory efficient, hashing scheme would also work. To index the 1-mismatch hits, PanArray adds each probe's $3k$ possible 1-mimsatch permutations to the index as well. The result of the indexing is a list of positions and segments for every possible probe of the pan-genome.

For CGH arrays, each probe is considered equivalent to its reverse complement, but for expression arrays, forward and reverse strand probes must be considered independently. Probe matches are listed on the strand on which they appear, therefore the sequence to be synthesized on the array may need to be reverse complemented. If desired, this final list of probe candidates can then be filtered based on typical criterion such as melting temperature, GC content, repeat content, etc. If the filtering is very aggressive, some segments may not contain any candidate probes and can be removed from further consideration.

2.2 Probe Selection

2.2.1 Naive Greedy Algorithm As detailed above, selecting a minimum probe set for tiling S is equivalent to finding the minimum hitting set of P. Let a segment hit by at least one probe be termed as *covered*. A naive greedy algorithm for finding a small hitting set H is to choose, for each uncovered segment, a probe hitting the segment that also hits the most other segments. The hope being that choosing probes with the most hits will minimize the total number of probes necessary to cover all segments. As before, S is the segmented pan-genome, and P_i is the subset of probes that hit the segment s_i. Let S_p be the subset of segments hit by probe p, and U be the set of currently uncovered segments.

Naive Greedy Algorithm

$H = \emptyset$
$U = S$
foreach $s_i \in S$
  if $s_i \in U$ **select** $\underset{p \in P_i}{\operatorname{argmax}} \left|S_p\right|$
    $U \leftarrow U - S_p$
    $H \leftarrow H \cup \{p\}$
return H

The result H is the list of probes that should be fabricated for the array. Note that if the number of probes in H is larger (or smaller) than desired, the fixed window size can be increased (or decreased) as necessary to adjust the density of the tiling. However, this algorithm fails to account for the fact that after each iteration, the effective coverage of the remaining probes may be reduced. This is because after selecting a probe p, all other probes that hit a segment in S_p will see their residual coverage reduced. Take for instance two probes that hit all the same segments. Choosing the first probe reduces the residual coverage of the second probe to zero.

2.2.2 Greedy PanArray Algorithm The residual compilation of the naive algorithm is avoided by the true greedy algorithm that reconsiders the effective, or residual, coverage of all probes at each iteration. The full greedy algorithm chooses, while uncovered segments remain, the probe that hits the most currently uncovered segments.

Greedy PanArray Algorithm

$H = \emptyset$
$U = S$
while $U \neq \emptyset$
  select $\underset{p \in P}{\operatorname{argmax}} \left|S_p \cap U\right|$
    $U \leftarrow U - S_p$
    $H \leftarrow H \cup \{p\}$
return H

As a downside, Greedy PanArray can be costly if the value of $|S_p \cap U|$ is recomputed for all S_p during each iteration. Because both $|P|$ and $|S|$ can be on the order of millions for a large pan-genome, recomputing the coverages during each iteration is infeasible. Thankfully, not all values have to be recomputed after each iteration. For some probe p and its corresponding segments S_p, let r_p equal the residual coverage of p after some other probes have already been selected ($r_p = |S_p \cap U|$). Note that for any p, its residual coverage r_p can never increase. After each iteration, a probe's coverage either remains the same or decreases because one of its segments was hit by the prior iteration. Therefore, instead of recomputing all residuals after each iteration, it is sufficient to maintain a priority queue of residual coverages and only update values at the front of the queue. After each iteration, all residual coverages are invalidated. During the next iteration, new r_p values are computed for the probes at the front of the queue and their values are reinserted in the queue. Often, newly computed residuals will quickly return to the head of the queue before all the invalid residuals have been recomputed. At this point it is unnecessary to update any other residuals because their new values cannot be greater than their current value. Therefore, the head of the queue must be the maximum updated residual. This general strategy is often called lazy evaluation because computations are preformed at the last minute on only the necessary elements. This avoids many unnecessary computations and drastically improves the performance of the algorithm.

2.3 Annotation

The flexibility of the PanArray design algorithm is founded on its probe-centric approach. Because it does not require any identification and clustering of genes, the design is independent of any genome annotation, and instead of building the annotation into the design of the array, the annotation can be mapped onto the array after the design. This strategy allows for unannotated genomes to be included on the array and annotation updates to be incorporated as they become available. If the tiling covers both strands of the genome, the chip can also be used to search for unknown transcripts.

Included with the final probe set H is the list of locations on the pan-genome that each probe matches. If the genome sequence is updated, the location information can be easily recovered by remapping the probes to the genome using a matching tool such as MUMmer (Delcher, et al., 1999; Delcher, et al., 2002; Kurtz, et al., 2004) or Vmatch (Kurtz, 2003). To annotate the array, probes are mapped to all the features who's location coincides their own. The result is a many-to-many mapping with each feature being targeted by multiple probes, and a single probe possibly targeting multiple features (e.g. conserved genes between strains).

3 RESULTS

The PanArray algorithm described above was used to designed a pan-genome tiling array for the species *Listeria monocytogenes*. The design was constructed for a 385,000 feature NimbleGen array with a probe length of 50 nt. All 20 *L. monocytogenes* genomes listed in Table 1 were included in the design, with a combined sequence length of 54,810,759 bp. To avoid tiling low quality or contaminant sequence, contigs less than 2 Kbp in length were discarded. Because the specificity of hybridization for a 50 nt probe cannot distinguish between a few mismatches, probes differing by a single mismatch were considered equivalent during the design phase. The segment length was set to 24 bp, which yields an expected probe coverage of about 50 / 24 = 2.08x. These parameters guarantee that every base-pair of the pan-genome will be covered by at least one probe, since the probe length is more than twice the segment length.

The *L. monocytogenes* pan-genome sequence was divided into approximately 2.3 million 24 bp segments. To hit each segment, the PanArray algorithm selected 373,389 distinct probes mapping to 2,893,387 positions in the pan-genome. On average, each probe in the design targets about 8 different positions in the pan-genome. Rather than being repeated sequences within the same genome, these different locations most often refer to a conserved locus in multiple strains (Figure 2). Interestingly, the degree of probe reuse corresponds well with the known evolutionary relationship of the strains. Included on the chip are 8 genomes from lineage I, 10 from

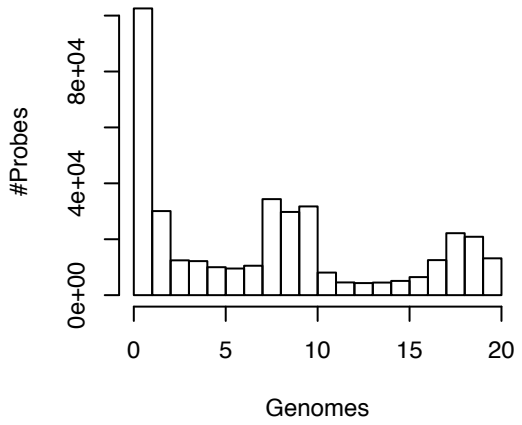

Fig. 2. Histogram of the number of *Listeria monocytogenes* genomes matching a single 50-mer PanArray probe with 0 or 1-mismatches.

lineage II, and 2 from lineage III. This would suggest that the peak at *Genomes* = 1 in Figure 2 is for strain-specific probes; the peaks around 2 and 9 are for lineage-specific probes; and the peak around 20 is for species-specific probes.

Because this is a dense tiling of the entire genome, it is unnecessary to optimize probes for uniqueness, as is done in standard expression arrays with only a few probes per gene. Nevertheless, it is recommended by array manufactures to avoid highly repetitive sequences. If necessary, such probes could be prescreened and discarded before running the PanArray algorithm. This may be necessary for microbial genomes with a large number of insertion elements, but the *L. monocytogenes* strains used for this array are not highly repetitive. The most repetitive probe used in the design targets a "cell wall surface anchor protein" family and occurs a maximum of 16 times on a single *L. monocytogenes* genome. In other cases, the relatively long 50 nt probe length assures that the majority of probes on the array match only a single location per genome.

To augment the original PanArray design, an additional 228 negative control probes were added to the array, chosen from *Bacillus spp*. which is a known cohabitant of *Listeria.* The negative control probes were chosen to be specific to *Bacillus spp.* using the Insignia genomic signature design pipeline (Phillippy, et al., 2007). The remaining 11,838 features on the array were filled by selecting individual probes to supplement the lowest coverage regions of the design. All probes were checked to conform with NimbleGen design restraints, and a few probes were trimmed to meet synthesis cycle limits. The resulting *L. monocytogenes* pan-genome array has an average coverage of 2.65x, with a median probe offset of 21 bp, and a modal offset equal to the segment length of 24 bp. The full distribution of probe offsets is given in Figure 3. The heavy left tail suggests this may be a non-optimal solution that is slightly denser than expected (2.65x vs. 2.08x expected). The majority of targeted sequences exactly match their probe (75%) and the remainder contain only a single mismatch (25%).

The performance gain of PanArray over more naive methods is significant. For instance, selecting a single probe from each window requires over 2.2 million probes, some of which may be exact duplicates. The slightly more principled Naive Greedy chooses a more reasonable 1,739,242 probes, but is still well over the 385,000 probe limit. The Greedy PanArray meets this limit and vastly outperforms the other methods—requiring only 373,389 probes to cover the entire pan-genome (Figure 4). Thanks to the lazy evaluation speedup, the PanArray algorithm is also comparable in runtime to the naive algorithm. On a single 2.4 GHz processor, the greedy algorithm *without* lazy evaluation never finished; the Naive Greedy algorithm took 29 s; and the Greedy PanArray algorithm took 130 s. Instead of optimization, the runtime for the entire design process is dominated by building the *k*-mer index, which required 84 m using the compressed keyword tree.

Analysis of aCGH experiments is usually conducted on signal ratios between a reference and experimental hybridization. Duplications or deletions in the experimental samples is evident as non-zero values of the log ratio of the two normalized signals. So called *segmentation* algorithms examine this log ratio across multiple positions in reference sequence to determine the boundaries of the variations (Olshen, et al., 2004; Willenbrock and Fridlyand, 2005). The most accurate methods consider not just individual probes, but also a context of probes from a similar genomic location. This adds robustness to the analysis, because a single low intensity probe is more likely to be an experimental error if it is flanked by high intensity probes. However, this analysis requires both a reference signal and a reference coordinate system on which the probes are tiled. In a pan-genome array, there is no single, linear reference genome and some probes will always be negative for a reference hybridization. This complicates segmentation analysis because the log ratio will be near zero when both strains *do* have a gene and when both strains *do not* have a gene.

For strain genotyping, it is more informative to know what genes from the pan-genome are present in the experimental strain, rather than the differences between a single reference and experimental strain. For gene-level analysis, direct analysis of the individual probe intensities provides comparable sensitivity and specificity versus segmentation analysis (Willenbrock, et al., 2007). A probe-based approach provides the most flexibility for pan-genome array analysis, because each probe can be individually scored based on its own intensity, and the genes can be classified based on the aggregated scores of the individual probe scores without the need for a control hybridization.

Emerging sequencing technologies promise to eventually replace

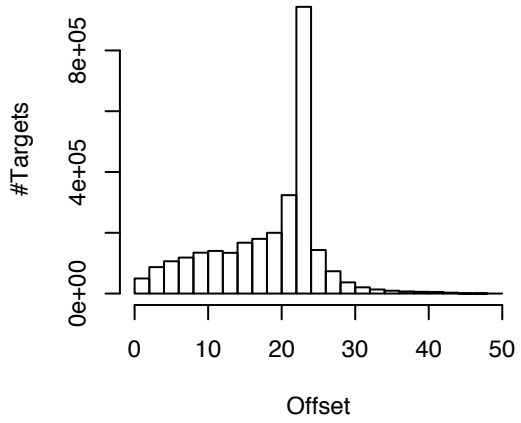

Fig. 3. Histogram of the offsets between adjacent probe targets in the *Listeria monocytogenes* pan-genome allowing for 0 or 1-mismatches to the probe.

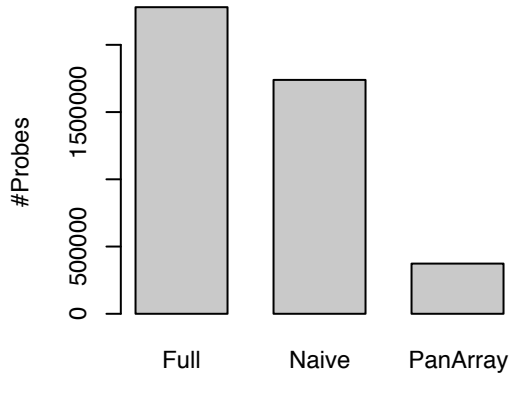

Fig. 4. Number of probes required to cover all 24 bp segments of the *Listeria monocytogenes* pan-genome using a full tiling, the Naive Greedy algorithm, and the Greedy PanArray algorithm.

aCGH with whole-genome sequencing. Until then, aCGH remains more economical for custom genotyping studies, such as this one, and copy number variation analysis (Shendure, 2008). Probe based methods, like microarray and PCR, are especially well suited for real-time pathogen detection, surveillance, and diagnostics, where a known sequence of DNA must be targeted from a vast environment (Phillippy, et al., 2007; Slezak, et al., 2003; Tembe, et al., 2007). For instance, a pan-genome array could be used for the detection and genotyping of pathogens from a large environment, without needing to isolate the individual cells. It could also be used to capture all species specific genomic material from an environment, which could then be directly processed or sequenced separately from the metagenome. Microarray based genomic capture has already been applied to targeted human resequencing as an efficient means of isolating desired sequencing templates (Albert, et al., 2007; Okou, et al., 2007; Porreca, et al., 2007).

PanArray is a novel and efficient algorithm for designing comprehensive pan-genome tiling arrays. Using a probe-centric design approach, functional clustering pitfalls are eliminated, and every unique sequence of the pan-genome is guaranteed to be included on the chip. The pan-genome array described here is the first of its kind and contains the entire genetic material for 20 distinct *L. monocytogenes* strains on a single array of 385,000 probes. The PanArray design software, and the *L. monocytogenes* array design and annotation, are freely available upon request.

REFERENCES

Albert, T.J., Molla, M.N., Muzny, D.M., Nazareth, L., Wheeler, D., Song, X., Richmond, T.A., Middle, C.M., Rodesch, M.J., Packard, C.J., Weinstock, G.M. and Gibbs, R.A. (2007) Direct selection of human genomic loci by microarray hybridization, *Nat Methods*, **4**, 903-905.

Bertone, P., Trifonov, V., Rozowsky, J.S., Schubert, F., Emanuelsson, O., Karro, J., Kao, M.Y., Snyder, M. and Gerstein, M. (2006) Design optimization methods for genomic DNA tiling arrays, *Genome Res*, **16**, 271-281.

Borucki, M.K., Kim, S.H., Call, D.R., Smole, S.C. and Pagotto, F. (2004) Selective discrimination of Listeria monocytogenes epidemic strains by a mixed-genome DNA microarray compared to discrimination by pulsed-field gel electrophoresis, ribotyping, and multilocus sequence typing, *J Clin Microbiol*, **42**, 5270-5276.

Call, D.R., Borucki, M.K. and Besser, T.E. (2003) Mixed-genome microarrays reveal multiple serotype and lineage-specific differences among strains of Listeria monocytogenes, *J Clin Microbiol*, **41**, 632-639.

Chung, W.H., Rhee, S.K., Wan, X.F., Bae, J.W., Quan, Z.X. and Park, Y.H. (2005) Design of long oligonucleotide probes for functional gene detection in a microbial community, *Bioinformatics*, **21**, 4092-4100.

Delcher, A.L., Kasif, S., Fleischmann, R.D., Peterson, J., White, O. and Salzberg, S.L. (1999) Alignment of whole genomes, *Nucleic Acids Res*, **27**, 2369-2376.

Delcher, A.L., Phillippy, A., Carlton, J. and Salzberg, S.L. (2002) Fast algorithms for large-scale genome alignment and comparison, *Nucleic Acids Res*, **30**, 2478-2483.

Doumith, M., Cazalet, C., Simoes, N., Frangeul, L., Jacquet, C., Kunst, F., Martin, P., Cossart, P., Glaser, P. and Buchrieser, C. (2004) New aspects regarding evolution and virulence of Listeria monocytogenes revealed by comparative genomics and DNA arrays, *Infect Immun*, **72**, 1072-1083.

Farber, J.M. and Peterkin, P.I. (1991) Listeria monocytogenes, a food-borne pathogen, *Microbiol Rev*, **55**, 476-511.

Feige, U. (1998) A threshold of ln n for approximating set cover, *Journal of the ACM (JACM)*, **45**, 634-652.

Feng, S. and Tillier, E.R. (2007) A fast and flexible approach to oligonucleotide probe design for genomes and gene families, *Bioinformatics*, **23**, 1195-1202.

Graf, S., Nielsen, F.G., Kurtz, S., Huynen, M.A., Birney, E., Stunnenberg, H. and Flicek, P. (2007) Optimized design and assessment of whole genome tiling arrays, *Bioinformatics*, **23**, i195-204.

Johnson, D. (1973) Approximation algorithms for combinatorial problems. ACM New York, NY, USA, 38-49.

Ksiazek, T.G., Erdman, D., Goldsmith, C.S., Zaki, S.R., Peret, T., Emery, S., Tong, S., Urbani, C., Comer, J.A., Lim, W., Rollin, P.E., Dowell, S.F., Ling, A.E., Humphrey, C.D., Shieh, W.J., Guarner, J., Paddock, C.D., Rota, P., Fields, B., DeRisi, J., Yang, J.Y., Cox, N., Hughes, J.M., LeDuc, J.W., Bellini, W.J. and Anderson, L.J. (2003) A novel coronavirus associated with severe acute respiratory syndrome, *N Engl J Med*, **348**, 1953-1966.

Kurtz, S. (2003) A Time and Space Efficient Algorithm for the Substring Matching Problem. Zentrum für Bioinformatik, Universität Hamburg.

Kurtz, S., Phillippy, A., Delcher, A.L., Smoot, M., Shumway, M., Antonescu, C. and Salzberg, S.L. (2004) Versatile and open software for comparing large genomes, *Genome Biol*, **5**, R12.

Medini, D., Donati, C., Tettelin, H., Masignani, V. and Rappuoli, R. (2005) The microbial pan-genome, *Curr Opin Genet Dev*, **15**, 589-594.

Mockler, T.C., Chan, S., Sundaresan, A., Chen, H., Jacobsen, S.E. and Ecker, J.R. (2005) Applications of DNA tiling arrays for whole-genome analysis, *Genomics*, **85**, 1-15.

Okou, D.T., Steinberg, K.M., Middle, C., Cutler, D.J., Albert, T.J. and Zwick, M.E. (2007) Microarray-based genomic selection for high-throughput resequencing, *Nat Methods*, **4**, 907-909.

Olshen, A.B., Venkatraman, E.S., Lucito, R. and Wigler, M. (2004) Circular binary segmentation for the analysis of array-based DNA copy number data, *Biostatistics*, **5**, 557-572.

Phillippy, A.M., Mason, J.A., Ayanbule, K., Sommer, D.D., Taviani, E., Huq, A., Colwell, R.R., Knight, I.T. and Salzberg, S.L. (2007) Comprehensive DNA signature discovery and validation, *PLoS Comput Biol*, **3**, e98.

Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W.L., Chen, C., Zhai, Y., Dairkee, S.H., Ljung, B.M., Gray, J.W. and Albertson, D.G. (1998) High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays, *Nat Genet*, **20**, 207-211.

Porreca, G.J., Zhang, K., Li, J.B., Xie, B., Austin, D., Vassallo, S.L., LeProust, E.M., Peck, B.J., Emig, C.J., Dahl, F., Gao, Y., Church, G.M. and Shendure, J. (2007) Multiplex amplification of large sets of human exons, *Nat Methods*, **4**, 931-936.

Shendure, J. (2008) The beginning of the end for microarrays?, *Nat Methods*, **5**, 585-587.

Slezak, T., Kuczmarski, T., Ott, L., Torres, C., Medeiros, D., Smith, J., Truitt, B., Mulakken, N., Lam, M., Vitalis, E., Zemla, A., Zhou, C.E. and Gardner, S. (2003) Comparative genomics tools applied to bioterrorism defence, *Brief Bioinform*, **4**, 133-149.

Tembe, W., Zavaljevski, N., Bode, E., Chase, C., Geyer, J., Wasieloski, L., Benson, G. and Reifman, J. (2007) Oligonucleotide fingerprint identification for microarray-based pathogen diagnostic assays, *Bioinformatics*, **23**, 5-13.

Tettelin, H., Masignani, V., Cieslewicz, M.J., Donati, C., Medini, D., Ward, N.L., Angiuoli, S.V., Crabtree, J., Jones, A.L., Durkin, A.S., Deboy, R.T., Davidsen, T.M., Mora, M., Scarselli, M., Margarit y Ros, I., Peterson, J.D., Hauser, C.R., Sundaram, J.P., Nelson, W.C., Madupu, R., Brinkac, L.M., Dodson, R.J., Rosovitz, M.J., Sullivan, S.A., Daugherty, S.C., Haft, D.H., Selengut, J., Gwinn, M.L., Zhou, L., Zafar, N., Khouri, H., Radune, D., Dimitrov, G., Watkins, K., O'Connor, K.J., Smith, S., Utterback, T.R., White, O., Rubens, C.E., Grandi, G., Madoff, L.C., Kasper, D.L., Telford, J.L., Wessels, M.R., Rappuoli, R. and Fraser, C.M. (2005) Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome", *Proc Natl Acad Sci U S A*, **102**, 13950-13955.

Volokhov, D., Rasooly, A., Chumakov, K. and Chizhikov, V. (2002) Identification of Listeria species by microarray-based assay, *J Clin Microbiol*, **40**, 4720-4728.

Wang, D., Coscoy, L., Zylberberg, M., Avila, P.C., Boushey, H.A., Ganem, D. and DeRisi, J.L. (2002) Microarray-based detection and genotyping of viral pathogens, *Proc Natl Acad Sci U S A*, **99**, 15687-15692.

Wang, D.G., Fan, J.B., Siao, C.J., Berno, A., Young, P., Sapolsky, R., Ghandour, G., Perkins, N., Winchester, E., Spencer, J., Kruglyak, L., Stein, L., Hsie, L., Topaloglou, T., Hubbell, E., Robinson, E., Mittmann, M., Morris, M.S., Shen, N., Kilburn, D., Rioux, J., Nusbaum, C., Rozen, S., Hudson, T.J., Lipshutz, R., Chee, M. and Lander, E.S. (1998) Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome, *Science*, **280**, 1077-1082.

Willenbrock, H. and Fridlyand, J. (2005) A comparison study: applying segmentation to array CGH data for downstream analyses, *Bioinformatics*, **21**, 4084-4091.

Willenbrock, H., Hallin, P.F., Wassenaar, T.M. and Ussery, D.W. (2007) Characterization of probiotic Escherichia coli isolates with a novel pan-genome microarray, *Genome Biol*, **8**, R267.

Zhang, C., Zhang, M., Ju, J., Nietfeldt, J., Wise, J., Terry, P.M., Olson, M., Kachman, S.D., Wiedmann, M., Samadpour, M. and Benson, A.K. (2003) Genome diversification in phylogenetic lineages I and II of Listeria monocytogenes: identification of segments unique to lineage II populations, *J Bacteriol*, **185**, 5573-5584.