Design of genome-wide HD-FISH probes

Magda Bienko, Nicola Crosetto, Leonid Teytelman, Sandy Klemm, Shalev Itzkovitz, and Alexander van Oudenaarden

Abstract

This protocol describes the design of primer pairs against the human genome for the synthesis of probes for high-definition DNA FISH (HD-FISH). This pipeline selects PCR primer pairs with optimal thermodynamic features, delimiting amplicons 200–220 nucleotides in length, and filters out primer pairs amplifying multiple targets as well as cross-hybridizing amplicons. Using such primers, highly specific double-stranded probes can be rapidly generated for virtually any desired genomic locus by fluorescently labeling pooled amplicons after PCR.

While this protocol describes the design against the human genome, we have also used it to geneate a genome-wide library for mouse. The design and method should work across other organisms as well.

For more information please see the full paper (and the dedicated hdfish.eu website): Bienko, Magda et al. "A Versatile Genome-Scale PCR-Based Pipeline for High-Definition DNA FISH." *Nature methods* 10.2 (2013): 122–124. *PMC*. Web. 9 Nov. 2015.

Citation: Magda Bienko, Nicola Crosetto, Leonid Teytelman, Sandy Klemm, Shalev Itzkovitz, and Alexander van Oudenaarden Design of genome-wide HD-FISH probes. **protocols.io**

dx.doi.org/10.17504/protocols.io.d6m9c5

Published: 09 Nov 2015

Guidelines

OVERVIEW

1. Each chromosome is split into 500-bp tiled fragments (sliding in 100bp steps)
2. Each fragment is searched against the human genome with blat*
3. Unique fragments are merged if overlapping
4. Primer 3 is run on the merged sequences for tiled primer pairs design.
5. Each primer pair is checked for uniqueness by e-PCR.
6. Check each probe for uniqueness against the genome with a more sensitive BLAT search.**

*First round of BLAT (on 500bp-windows) uses 80% cutoff for
matches.
**Final round of BLAT (on the 200bp probes) uses 70% cutoff.

SOFTWARE

Homology searches are done with BLAT (http://genome.ucsc.edu/FAQ/FAQblat.html)

Primers are designed with "Primer3"
(http://primer3.sourceforge.net/)

In silico PCR is done with "e-PCR"
(http://www.ncbi.nlm.nih.gov/tools/epcr/)

Total sequence for primer design, with single-hits only:

1,228,290,200bp (step 3)
5,120,725 initial primers designed (step 4)
5,080,020 after in silico PCR (step 5)
4,823,784 final probes
-Final round of BLAT (on the 200bp probes) uses 70% cutoff.

- e-PCR search done with maximum of 2bp mismatch in the primers.

NOTES

The pipeline uses a cluster, so many of the commands are specific to breaking up the analysis files and submitting parallel jobs to the cluster nodes.

All scripts are available at hdfish.edu.

ANALYSIS

Want to get a feel for the coverage of chromosomes.

- How many probes per 10KB?
- How many deserts with very few probes?
- Excluding the deserts, what is the probe density?

Below are the SQL analyses to answer the above.

//
-Load the final probe positions into mySQL (human DB)

```
CREATE TABLE probes(
  chrom VARCHAR(20),
   start INT,
   stop INT
) ;
```

```
LOAD DATA LOCAL INFILE '/home/lenny/projects/hd_fish/db/round2_unique_probes.tbl' INTO TABLE probes;
```

-Count "N"s in the fasta file
(on rous)

```
cd /home/nylenny/hdfish/databases/fasta/
```

```
for file in chr*.fa; do cat $file | perl -ne 'chomp;if (m/^>/){$chrom=$_;$chrom=~s/>//;print "$chrom\t";}else{$n_count = tr/nN//;$totaln+=$n_count}END{print "$totaln\n";}' >>
```

```
human_chr_ncounts.txt ; done

CREATE TABLE chroms(
   chrom VARCHAR(20),
   size INT
) ;

CREATE TABLE ncounts(
   chrom VARCHAR(20),
   totalN INT
) ;

LOAD DATA LOCAL INFILE '/home/lenny/projects/hd_fish/db/human_chr_ncounts.tbl' INTO TABLE
ncounts;

create table chroms_full select chroms.chrom,chroms.size,ncounts.totalN n_count, size-totalN
adjusted_size from chroms, ncounts where chroms.chrom=ncounts.chrom;

drop table chroms;
drop table ncounts;
alter table chroms_full rename to chroms;

- Count probes/10KB for each chromosome

CREATE TABLE temp (select chrom,count(*) total_probes from probes group by chrom);

create table probes_bychrom select pb.chrom,pb.total_probes,chroms.size chrom_size,
round(10000*pb.total_probes/chroms.size,1) probes_per10kb from temp pb, chroms WHERE
pb.chrom=chroms.chrom order by chroms.size DESC;

create table probes_bychrom_adj select pb.chrom,pb.total_probes,chroms.adjusted_size
adj_chrom_size, round(10000*pb.total_probes/chroms.adjusted_size,1) probes_per10kb from temp pb,
chroms WHERE pb.chrom=chroms.chrom order by chroms.size DESC;

drop table temp;

- Look at 10KB-windows
create table windows select chrom,round(start/10000) start,count(*) total from probes group by
chrom,round(start/10000);
```

Protocol

BLAT database setup

Step 1.

Download the human genome

cmd COMMAND

```
cd /home/nylenny/hdfish/databases/fasta
```

```
ftp://hgdownload.cse.ucsc.edu/goldenPath/currentGenomes/Homo_sapiens/bigZips/chromFa.tar.gz
```

BLAT database setup

Step 2.

Remove all strange .fa files accept for chr1.fa chr2.fa...

cmd COMMAND

```
rm chrUn_gl0002* *_gl* *hap*
```

BLAT database setup

Step 3.

Make the 2bit file for BLAT

cmd COMMAND

```
for file in *.fa
     do qsub -b y /home/nylenny/programs/blat/faToTwoBit -
noMask /home/nylenny/hdfish/databases/fasta/$file /home/nylenny/hdfish/databases/blatdb/$fi
le.2bit
done
```

NOTES

Lenny Teytelman 09 Nov 2015

"qsub" is a command for submitting jobs to cluster nodes using the PBS Pro software (Portable Batch System).

BLAT database setup

Step 4.

Make an 11.ooc file

cmd COMMAND

```
cd /home/nylenny/hdfish/databases/blatdb
ls -1 *.2bit > human_genome_chrom.list
~/programs/blat/blat -makeOoc=11.ooc -
tileSize=11 human_genome_chrom.list /dev/null /dev/null
```

Split chroms into 500-bp fragments

Step 5.

Make a fasta file of 500-bp windows, in 100bp slidingsteps for each chromosome.

cmd COMMAND

```
cd /home/nylenny/hdfish/databases/fasta

for file in *.fa
do
     export infile=/home/nylenny/hdfish/databases/fasta/$file
     export outfile=/home/nylenny/hdfish/databases/split_chroms/${file%%.fa}._windows.fa
     qsub -o /home/nylenny/hdfish/cluster_eo/ -e  /home/nylenny/hdfish/cluster_eo/ -
V /home/nylenny/hdfish/qsub_runs/split_chroms.sh
done
```

Split chroms into 500-bp fragments

Step 6.

Get human chromosome sizes

cmd COMMAND

```
for file in *.fa;do echo $file; grep -v '>' $file | wc;done > human_chr_sizes.txt

cat human_chr_sizes.txt | perl -
ne 'chomp;if(m/fa$/){s/.fa//;print "$_\t";}else{s/^\s*//;my($lines,$words,$total_chars)=spl
it(" +");$chrom_size=$total_chars-$lines;print "$chrom_size\n";}' > human_chr_sizes.txt2

mv human_chr_sizes.txt2 human_chr_sizes.txt
```

NOTES

Lenny Teytelman 09 Nov 2015

Steps 6-8 are a sanity check. Making sure that the resulting split_windows chrom sizes match the actual size of the human chromosomes in the initial .fasta files.

Split chroms into 500-bp fragments

Step 7.

Get the sizes of the split chroms

cmd COMMAND

```
cd /home/nylenny/hdfish/databases/split_chroms

for file in *.fa; do tail -2 $file | head -1; done | tr -d '>' |tr '_' '\t' | cut -
f 1,3|sort > split_chrom_sizes.txt
```

Split chroms into 500-bp fragments

Step 8.

Visually inspect the output from this step to compare real chroms against the split ones

cmd COMMAND

```
join -j 1 ../fasta/human_chr_sizes.txt split_chrom_sizes.txt | tr ' ' '\t' | perl -
ne 'chomp;my($chrom,$real_size,$split_size)=split("\t");print "$chrom\t",$real_size-
$split_size,"\n";'
```

BLAT, first round

Step 9.

BLAT the split windows for each chromosome against the entire human genome, one chromosome at a time.

cmd COMMAND

```
export windowsfile=chr1._windows.fa

cd /home/nylenny/hdfish/databases/blatdb

for chromosome in *.2bit
do
    export chrom=$chromosome
    qsub -V -o /home/nylenny/hdfish/cluster_eo/${windowsfile}_to_$chrom.o -
e /home/nylenny/hdfish/cluster_eo/${windowsfile}_to_$chrom.e /home/nylenny/hdfish/qsub_runs
/blat_subm_script.sh
done
```

NOTES

Lenny Teytelman 09 Nov 2015

"qsub" is a command for submitting jobs to cluster nodes using the PBS Pro software (Portable Batch System).

Lenny Teytelman 09 Nov 2015

This first round of BLAT (on 500bp-windows) is more permissive and uses 80% cutoff for matches.

BLAT, first round

Step 10.

Check for completion of the blat against each chromosome. (The last blat hit should be close to the end of the query file.)

cmd COMMAND

```
cd /home/nylenny/hdfish/blat_results

for file in chr3*; do echo $file |perl -
ne 'chomp;s/chr4._windows.fa_to_//;s/.fa.2bit.psl//;print "$_\t";'; tail -1 $file | cut -
f 10 ;done

for file in chr*; do echo $file |perl -
```

```
ne 'chomp;s/chr5._windows.fa_to_//;s/.fa.2bit.psl//;print "$_\t";'; tail -1 $file | cut -
f 10 ;done > sizes.txt
```

Find unique fragments

Step 11.

Merge all the .psl files for each chromosome

cmd COMMAND

```
qrsh

cd /home/nylenny/hdfish/blat_results
mkdir by_chrom
mv *.psl by_chrom/

for file in chr1 chr10 chr11 chr12 chr13 chr14 chr15 chr16 chr17 chr18 chr19 chr2 chr20 chr
21 chr22 chr3 chr4 chr5 chr6 chr7 chr8 chr9 chrX chrY
do
  qsub -e /home/nylenny/hdfish/cluster_eo/ -
o /home/nylenny/hdfish/blat_results/$file.psl  -
b y  cat /home/nylenny/hdfish/blat_results/by_chrom/${file}._*.psl
done

for file in chr11 chr12 chr13 chr14 chr15 chr16 chr17 chr18 chr19 chr2 chr20 chr21 chr22 ch
r3 chr4 chr5 chr6 chr7 chr8
```

NOTES

Lenny Teytelman 09 Nov 2015

In this section, we are processing the BLAT results to identify all 500-bp fragments that have a single hit to the genome.

Find unique fragments

Step 12.

Count total BLAT hits per fragment

cmd COMMAND

```
cd /home/nylenny/hdfish/blat_results/

for file in *.psl
do
     export infile=/home/nylenny/hdfish/blat_results/$file
     export outfile=/home/nylenny/hdfish/blat_results/counts/${file%%.psl}.blatcounts.txt
     qsub -o /home/nylenny/hdfish/cluster_eo/ -e  /home/nylenny/hdfish/cluster_eo/ -
V /home/nylenny/hdfish/qsub_runs/count_blat_hits.sh
done
```

Merge unique fragments

Step 13.

Sort by positional start

cmd COMMAND

```
qrsh
cd /home/nylenny/hdfish/blat_results/counts

for file in *.txt
do
   tr '_' '\t' < $file  | cut -f 2,4 | sort -
k 1n > ${file%%.blatcounts.txt}_sorted_blatcounts.csv
done
```

NOTES

Lenny Teytelman 09 Nov 2015

In this section, we merge all overlapping single-hit windows and get their DNA sequences.

Merge unique fragments

Step 14.

Make a list of windows with just a single hit

cmd COMMAND

```
for file in *.csv
do
 cat $file | perl -
ne 'chomp;my($start,$count)=split("\t");print "spacer\tspacer\t$start\t",$start+499,"\n" if
 $count
```

Merge unique fragments

Step 15.

Merge the overlapping windows

cmd COMMAND

```
cd /home/nylenny/hdfish/blat_results/singles

for file in *.txt
do
    perl /home/nylenny/hdfish/scripts/interval_merger.pl  -
f $file  > ${file%%singlehit_windows.txt}_singles_merged.txt
done
```

NOTES

Lenny Teytelman 09 Nov 2015

[We're now done with the round1 blat .psl output files. Can zip them to save space.]

cd /home/nylenny/hdfish/blat_results/

for file in *.psl
do
qsub -e /home/nylenny/hdfish/cluster_eo/ -o /home/nylenny/hdfish/cluster_eo/ -b y gzip /home/nylenny/hdfish/blat_results/$file
done

Merge unique fragments

Step 16.

On local machine, make a fasta file with the merged unique sequences

cmd COMMAND

```
cd /media/Elements/projects/hdfish/unique_regions

for file in *.txt; do perl ~/projects/hd_fish/scripts/name_intervals.pl -
i  $file > ${file%%_singles_merged.txt}_unique_toget.txt;done

for file in *toget.txt; do chrom=${file%%_unique_toget.txt}; perl ~/programs/Scripts/get_fa
sta_sequence.pl -f /media/Elements/projects/hdfish/databases/fastas/human/${chrom}.fa -
seq_list $file > /media/Elements/projects/hdfish/databases/fastas/unique_regions/${chrom}_u
nique.fa; done
```

Design tiled primer pairs for PCR

Step 17.

Convert the fasta sequences to primer3 input records

cmd COMMAND

```
cd /media/Elements/projects/hdfish/databases/fastas/unique_regions/

for file in *.fa
do
```

```
    perl ~/projects/hd_fish/scripts/make_primer3_records.pl  -
i $file > /media/Elements/projects/hdfish/databases/primer3/${file%%_unique.fa}_unique_for_
primer3.txt
done
```

NOTES

Lenny Teytelman 09 Nov 2015

In this section, we run Primer3 to design tiled PCR primer pairs against the unique genome sections identified above.

Design tiled primer pairs for PCR

Step 18.

Transfer all the files to the cluster: rous:/home/nylenny/hdfish/databases/primer3

Design tiled primer pairs for PCR

Step 19.

Run Primer3 on each chrom_unique file

cmd COMMAND

```
cd /home/nylenny/hdfish/databases/primer3

for file in *.txt
do
     export infile=/home/nylenny/hdfish/databases/primer3/$file
     export outfile=/home/nylenny/hdfish/primers/primer3_output/${file%%_unique_for_primer3
.txt}_primer3_output.txt
     qsub -o /home/nylenny/hdfish/cluster_eo/ -e  /home/nylenny/hdfish/cluster_eo/ -
V /home/nylenny/hdfish/qsub_runs/primer3_jobs.sh
done
```

Select unique primer pairs

Step 20.

Process the Primer3 output.

cmd COMMAND

```
cd /home/nylenny/hdfish/primers/primer3_output
for file in *.txt
do
   perl /home/nylenny/hdfish/scripts/parse_primer3.pl -i $file > $file.parsed
done
```

NOTES

Lenny Teytelman 09 Nov 2015

In this section, we use e-PCR to ensure that each primer pair has a unique match in the genome and will not cross-hybridize.

Select unique primer pairs

Step 21.

Count primers per chromosome (normalized to 10kb)

cmd COMMAND

```
wc *.parsed | tr -s ' ' '\t'  | grep -v total | cut -f 2,5 | cut -d '_' -f 1 |tr -
d 'chr' | sort -k 2n > primer_pair_counts.tmp


cat /home/nylenny/hdfish/databases/fasta/human_chr_sizes.txt |grep -v chrM | tr -
d 'chr' | sort -k 1n > human_sizes.tmp


join -1 2 -2 1 primer_pair_counts.tmp  human_sizes.tmp   | tr -s ' ' '\t' | perl -
ne 'chomp;my($chrom,$primers,$size)=split("\t");print "$chrom\t$primers\t$size\t",int(10000
0*$primers/$size)/10,"\n";' > primers_per10kb.txt
```

Select unique primer pairs

Step 22.

Make .sts files for e-PCR

cmd COMMAND

```
cd /home/nylenny/hdfish/primers/primer3_output

for file in *.parsed
do
 cat $file |  perl -
ne 'chomp;my($id,$leftp,$rightp,$left_start,$right_start)=split("\t");my($window,$wstart,$w
stop)=split("_",$id);print "$id\t$leftp\t$rightp\t",$wstart+$left_start,"\t",$wstart+$right
_start,"\n";' | sort -
k 4n > /home/nylenny/hdfish/primers/epcr_input/${file%%_primer3_output.txt.parsed}_primer3_
rawcoord.txt
done

cd /home/nylenny/hdfish/primers/epcr_input/

for file in *.txt
do
  cat $file |   perl -ne 'chomp;my($id,$
```

Select unique primer pairs

Step 23.

Run e-PCR against the entire human genome

cmd COMMAND

```
cd /home/nylenny/hdfish/primers/epcr_input


export stsfile=chr22_primer3_rawcoord.sts

cd /home/nylenny/hdfish/databases/fasta

for chromosome in *.fa
do
    export chrom=$chromosome
    qsub -V -
o /home/nylenny/hdfish/primers/epcr_output/${stsfile%%_primer3_rawcoord.sts}_to_$chrom.epcr
 -
e /home/nylenny/hdfish/cluster_eo/${stsfile}_to_$chrom.e /home/nylenny/hdfish/qsub_runs/epc
r.sh
done
```

Select unique primer pairs

Step 24.

Check for completion of batch jobs for each chromosome

cmd COMMAND

```
cd /home/nylenny/hdfish/primers/epcr_output

for file in chr22_*; do echo $file |perl -
ne 'chomp;s/chr3_to_//s;s/.fa.epcr//;print "$_\t";'; tail -1 $file ;done | tr -
s ' ' '\t' | cut -f 1,4


qrsh

cd /home/nylenny/hdfish/primers/epcr_output/
```

```
mkdir bychrom
mv * bychrom/

for file in chr1 chr10 chr11 chr12 chr13 chr14 chr15 chr16 chr17 chr18 chr19 chr2 chr20 chr
21 chr22 chr3 chr4 chr5 chr6 chr7 chr8 chr9 chrX chrY
do
  qsub -e /home/nylenny/hdfish/cluster_eo/ -o /home/nylenny/hdfish/primers/epcr_output/$fi
```

Select unique primer pairs

Step 25.

Exclude primers where e-PCR match is not to the exact same position as targeted

cmd COMMAND

```
cat chr*.txt| tr -s ' ' '\t' | tr -s '.' '\t' | tr -s ',' '\t' | tr -s '_' '\t' | perl -
ne 'chomp;$line=$_;my($echrom,$estart,$estop,$chrom,undef,undef,$start,$stop)=split("\t",$l
ine);if ($echrom ne $chrom or $start!=$estart or $stop!=$estop){print "$chrom\t$start\t$sto
p\n";}' |uniq|sort|uniq > wrong_match.txt

 perl /home/nylenny/hdfish/scripts/exclude_primers_bylist.pl -
i ../epcr_input/all_primers.txt -e wrong_match.txt > unique_primers.txt
```

BLAT, round 2

Step 26.

Now screen the probe sequence of the unique primers pairs by BLAT for uniqueness in the genome.

cmd COMMAND

```
cd /media/Elements/projects/hdfish/unique_primers

cat unique_primers.txt | tr -s '_' '\t' | cut -f 1,6,7 > unique_primers_toget.txt

perl ~/programs/Scripts/get_fasta_sequence.pl -
f /media/Elements/projects/hdfish/databases/fastas/human/all_human.fasta -
seq_list unique_primers_toget.txt > unique_probes_for_round2.fa

scp  unique_probes_for_round2.fa nylenny@rous.mit.edu:/home/nylenny/hdfish/databases/unique
_for_round2/


cd /home/nylenny/hdfish/databases/blatdb

for chromosome in *.2bit
do
    export chro
```

NOTES

Lenny Teytelman 09 Nov 2015

This second round of BLAT (on the 200bp probes) is stricter and uses 70% cutoff to avoid cross-hybridizing probes.

BLAT, round 2

Step 27.

Count hits per probe and select single-hit ones only

cmd COMMAND

```
cd hdfish/blat_results/round2/

cat probes_round2_to_chr*.psl > round2_to_allhuman.psl

perl /home/nylenny/hdfish/scripts/count_blat_hits.pl -
i round2_to_allhuman.psl > counts/round2_all_blatcounts.txt

cat round2_all_blatcounts.txt | perl -
```

```
ne 'chomp;my($name,$count)=split("\t");print "$name\t$count\n" if $count==1;' > round2_uniq
ue_probes.txt
```

ANNOTATIONS

Lenny Teytelman 10 Apr 2017