ARTIC primer scheme specification v3.0.0-alpha

Christopher Kent[1,2], and **Bede Constantinides**[1,2]

[1]The ARTICnetwork Collaborative Award, [2]Institute of Microbiology and Infection, University of Birmingham, Birmingham, UK.

Abstract

Polymerase chain reaction (PCR) followed by amplicon DNA sequencing enables fast, sensitive and cost-effective molecular characterisation of target genes and genomes. PCR involves the selective amplification of a target genomic region (amplicons) using pairs of single-stranded oligonucleotide primers, complementary to opposing strands flanking the target region. Multiple regions can be simultaneously amplified in a single reaction via multiplexed PCR, with multiple reactions enabling tiling amplicon sequencing (ARTIC sequencing), facilitating efficient enrichment of entire microbial genomes for whole genome sequencing. However, accurately reproducing a primer scheme and the corresponding bioinformatic analysis of amplicon sequencing data depends on knowledge of primer sequences, amplicon layout, and their coordinates with respect to a reference sequence. Analysis and reuse of amplicon sequencing data is currently hindered by the lack of a clearly defined data interchange format for primer scheme definitions, a problem highlighted by the proliferation of SARS-CoV-2 primer schemes during the COVID-19 pandemic. Here, we describe a text-based specification for describing primer sequences and locations with respect to a reference sequence. This specification formalises and expands on the existing interchange format initially used in the PrimalScheme primer design tool, and since adopted by a growing ecosystem of tooling. This specification designates the use of a primer.bed file, based on the Browser Extensible Data (BED) text format, and an accompanying reference.fasta text file for defining primer schemes, and probe-based qPCR assays. This specification is intended to facilitate the exchange of primer scheme definitions for oligonucleotide synthesis, wet-lab and bioinformatic analysis use cases.

Keywords Data standards, Primer Schemes, Amplicon Sequencing

Contents

1. primer.bed file

A primer.bed file describes a primer scheme in machine and human readable tabular format. Together with an accompanying reference.fasta, its purpose is to encapsulate all of the information needed to *i)* acquire the primers from suppliers or custom oligonucleotide synthesis, *ii)* combine the primers correctly to reproduce a pooled primer scheme, and *iii)* facilitate correct and reproducible bioinformatic analysis of resulting sequencing data. It therefore incorporates both wet lab and analytical elements. This information includes primer sequences, primer pools, coordinates and orientation with respect to a reference sequence, and optionally relative primer concentrations.

1.1. Format overview

`primer.bed` files are tab-delimited ASCII text files. Each line can either represent a *comment line* (prefixed with `#`) or a *record line* (`BedLine`), representing a single unique oligonucleotide primer or probe associated with an amplicon. An amplicon comprises at least two primer record lines each describing primers on different strands. A compliant `primer.bed` file contains one or more amplicons.

The format of `primer.bed` is based on Browser Extensible Data (BED) specification, with each oligonucleotide being treated as a genomic region, enabling compatibility with common BED file tooling.

1.2. Comment Line

Comment lines are minimally parsed, but can optionally contain a scheme-level (key, value) pair. To this end, comment lines containing a single “`=`” will be split, with the left and right sides representing a scheme-level key and value respectively.

1.3. record line (BedLine) field descriptions

Column	Field name	Type	Brief description	Restrictions
1	chrom	String	Chromosome name	`[A-Za-z0-9_]`
2	primerStart	Integer	Primer start position (zero-based, half-open)	Positive integer (`u64`)
3	primerEnd	Integer	Primer end position (zero-based, half-open)	Positive integer (`u64`)
4	primerName	String	Primer name	`[a-zA-Z0-9\-]+_[0-9]+_(LEFT|RIGHT|PROBE)_[0-9]+`
5	pool	Integer	Primer pool	Positive integer (`u64`)
6	strand	String	Primer strand	`[-+]`
7	primerSeq	String	The nucleotide sequence in 5′⟶ 3′	ASCII non-whitespace characters
8	primerAttributes	Optional(String)	List of record-level (key, value) pairs separated by `;`. e.g. `k1=v1;k2=v2`	ASCII non-whitespace characters

Table 1: The column structure and description of a BedLine

1.3.1. `chrom`

The name of the corresponding reference sequence chromosome for the primer. This must match a valid sequence ID inside an accompanying reference sequence FASTA file, by convention named `reference.fasta`.

1.3.2. `primerStart`

The start position of the primer on the `chrom` using BED-like zero-based, half-open coordinates.

1.3.3. `primerEnd`

The non-inclusive end position of the primer on the `chrom` using BED-like zero-based, half-open coordinates. Must be greater than `primerStart`.

1.3.4. `primerName`

The name of the primer in the form “`{prefix}_{ampliconNumber}_{class}_{primerNumber}`”.

- `prefix`: Must match regex `[a-zA-Z0-9\-]`. See best practices
- `ampliconNumber`: The number of the amplicon for its relevant `chrom`. Must be a positive integer incrementing from 1.
- `primerClass`: The class of the primer. Must be either `LEFT`, `RIGHT` or `PROBE`.
- `primerNumber`: The number of the primer. Must be a positive integer incrementing from 1.

1.3.5. `pool`

The PCR pool the primer belongs to. Must be a positive integer incrementing from 1[1].

1.3.6. `strand`

The strand of the primer must be either `+` or `-`. It must correspond to the `class` component of the `primerName` (see the description of `primerName` above). `LEFT` and `RIGHT` primers must be `+` and `-` respectively, while `PROBE` can be either.

1.3.7. `primerSeq`

The sequence of the primer in the 5′ to 3′ direction. Unrestricted to contain any non-whitespace ASCII character[2].

1.3.8. `primerAttributes`

An **optional** list of a (key, value) pairs used to denote additional arbitrary primer attributes, in the form of `pw=1.0;ps=10.0`. This is intentionally flexible to allow the storage of additional information. In a primer.bed file this can be represented as either an empty 8th column or only 7 columns.

1.3.8.1. Reserved keys

- `pw`: primerWeight. The concentration of individual primers can be altered to balance amplicon performance. Primer concentration in the PCR should be scaled by `primerWeight * [typical PCR conc]`. This is restricted to positive floating point numbers (`f64 > 0`).

1.4. Examples

1.4.1. Simple example

A seven column `primer.bed` file, with no `primerAttributes` or `comment lines`.

[1] “Existing schemes/literature use refer to `pool 1 and pool 2`. Therefore 1-based indexing is expected”

[2] “This is intentionally unrestricted (rather than IUPAC-only) to allow Primer Modification. Such as `/56-FAM/{primerSeq}` to represent 5′ 6-FAM fluorescent dye labelled probe”

```
MN908947.3  100 131 example_1_LEFT_1  1 + CTCTTGTAGATCTGTTCTCTAAACGAACTTT
MN908947.3  419 447 example_1_RIGHT_1 1 - AAAACGCCTTTTTCAACTTCTACTAAGC
MN908947.3  344 366 example_2_LEFT_1  2 + TCGTACGTGGCTTTGGAGACTC
MN908947.3  707 732 example_2_RIGHT_1 2 - TCTTCATAAGGATCAGTGCCAAGCT
```

1.4.2. Complex example

An eight column `primer.bed` file. With `primerAttributes` defined, and `comment lines` providing a `chrom` alias and explaining the `gc` `primerAttributes`.

```
# example scheme
# gc=fraction gc
# MN908947.3=sars-cov-2
MN908947.3  100 131 example_1_LEFT_1  1 + CTCTTGTAGATCTGTTCTCTAAACGAACTTT pw=1.4;gc=0.35
MN908947.3  419 447 example_1_RIGHT_1 1 - AAAACGCCTTTTTCAACTTCTACTAAGC  pw=1.4;gc=0.36
MN908947.3  344 366 example_2_LEFT_1  2 + TCGTACGTGGCTTTGGAGACTC  pw=1;gc=0.55
MN908947.3  707 732 example_2_RIGHT_1 2 - TCTTCATAAGGATCAGTGCCAAGCT pw=1;gc=0.44
```

1.4.3. qPCR example

An eight column `primer.bed` file. Showing a fictional qPCR assay. The specific dyes and quenchers are (optionally) included in the comments lines.

```
# example multiplexed-qPCR assay
# gc=fraction gc
# /3BHQ_1/=Black Hole Quencher 1
# /56-FAM/=FAM
# /5HEX/=HEX
target1  2010 2030 iad3_1_LEFT_1  1 + AAAGGTCAGTCAACCCGTTC pw=1
target1  2035 2060 iad3_1_PROBE_1  1 - /56-FAM/GCGTTGTTCAATTGCCTTGCTGATT/3BHQ_1/  pw=19.1
target1  2903 2923 iad3_1_RIGHT_1  1 - TCGGGCCACCGCGTATGAAG  pw=1
target2  5167 5187 rfw1_1_LEFT_1  1 + TCGTAGCATGGACTCGATGA pw=1
target2  5271 5296 rfw1_1_PROBE_1 1 + /5HEX/TGATCCGCGTTTACTGTTCGACGCG/3BHQ_1/  pw=20.2
target2  5301 5321 rfw1_1_RIGHT_1 1 - GTTTACCAAGGAACCATCCA  pw=1
```

1.5. Best Practices

`primer.bed` contain information about how to replicate the primer pools used in multiplexed PCR. They do not contain information about the PCR protocol, input material, or sequencing method and analysis. Therefore, additional information is needed for true reproducibility.

1.5.1. Other metadata standards

To explicitly differentiate different versions of `primer.bed`, this spec is designed to fit into larger metadata standards, such as primal-page with PrimalScheme Labs or primaschema with pha4ge primer-schemes

1.5.2. Other tooling

`primalbedtools` is a python package that carries out schema validation and conversion, and common operations on `primer.bed` files.

1.5.3. `primerName:prefix`

The `primerName:prefix` should be as unique as possible (for example a short uuid, `359ba5`) and different for each `chrom` and each scheme generation run.

- Using `primerName:prefix` like `scheme` or `sars-cov-2` might seen easier, however, will result in a freezer / LIMS full of similar names leading to pooling mistakes.

1.5.4. comment line

The `comment line`’s `key=value` pattern undergoes limited validation, and therefore should be not be critical for tooling. A suitable use case might be to document custom `primerAttributes`. Another use is providing aliases for different `chrom`.

2. reference.fasta file

A `reference.fasta` file contains the DNA sequences of all the primary-reference genomes, used in primer scheme generation. Its purpose is to provide a reference genome and coordinate system for use in reference-based assembly and consensus generation.

2.1. Format overview

`reference.fasta` files are typical ASCII-encoded `.fasta` format files, with text representing the nucleotide sequence of the reference. Each genome starts with a header line (starting with `>`) that denotes the id of the genome, followed by lines of nucleotide data.

All `chrom` fields of the record lines must have a corresponding id in the `reference.fasta`.

2.2. Examples

2.2.1. Single fasta

```
>MN908947.3
ATTAAAGGTTTATACCTTCCCA...
```

The corresponding `primer.bed` file should contain the `chrom` `MN908947.3`

2.2.2. Multi fasta

```
>MN908947.3
ATTAAAGGTTTATACCTTCCCA...
>NC_006432.1
CGGACACACAAAAAGAAAGAAA...
```

The corresponding `primer.bed` file should contain the `chrom` `MN908947.3` and `NC_006432.1`

2.3. Best practices

As the `reference.fasta` is often used for referenced-based assembly, using high quality genome with few `N`s or ambiguous bases is advisable. Using RNA sequences in the `reference.fasta` is not recommended, as DNA is expected.