

 PacificBiosciences / kineticsTools Watch 22  Star 5  Fork 4<> Code  Issues 14  Pull requests 1  Projects 0 Insights ▾

Branch: master ▾ kineticsTools / doc / manual.rst

Find file Copy path

 natechols note about pbsmrtpipe pipelines

c58e338 on Mar 23

1 contributor

227 lines (169 sloc) 13.6 KB

Raw Blame History  

Overview

kineticsTool loads IPDs observed at each position in the genome, and compares those IPDs to value expected for unmodified DNA, and outputs the result of this statistical test. The expected IPD value for unmodified DNA can come from either an *in-silico control* or an *amplified control*. The in silico control is trained by PacBio and shipped with the package. It predicts the IPD using the local sequence context around the current position. An amplified control dataset is generated by sequencing unmodified DNA with the same sequence as the test sample. An amplified control sample is usually generated by whole-genome amplification of the original sample.

Dependencies

kineticsTools depends on:

- pbcore (<http://github.com/PacificBiosciences/pbcore>)
- h5py
- numpy
- scipy

Modification Detection

The basic mode of kineticsTools does an independent comparison of IPDs at each position on the genome, for each strand, and emits various statistics to CSV and GFF (after applying a significance filter).

Modifications Identification

kineticsTools also has a Modification Identification mode that can decode multi-site IPD 'fingerprints' into a reduced set of calls of specific modifications. This feature has the following benefits:

- Different modifications occurring on the same base can be distinguished (for example m5C and m4C)
- The signal from one modification is combined into one statistic, improving sensitivity, removing extra peaks, and correctly centering the call

Algorithm

Synthetic Control

Studies of the relationship between IPD and sequence context reveal that most of the variation in mean IPD across a genome can be predicted from a 12-base sequence context surrounding the active site of the DNA polymerase. The bounds of the relevant context window correspond to the window of DNA in contact with the polymerase, as seen in DNA/polymerase

crystal structures. To simplify the process of finding DNA modifications with PacBio data, the tool includes a pre-trained lookup table mapping 12-mer DNA sequences to mean IPDs observed in C2 chemistry.

Filtering and Trimming

kineticsTools uses the Mapping QV generated by BLASR and stored in the cmp.h5 or BAM file (or AlignmentSet) to ignore reads that aren't confidently mapped. The default minimum Mapping QV required is 10, implying that BLASR has 90% confidence that the read is correctly mapped. Because of the range of readlengths inherent in PacBio data this can be changed in using the `--mapQvThreshold` command line argument, or via the SMRTPortal configuration dialog for Modification Detection.

There are a few features of PacBio data that require special attention in order to achieve good modification detection performance. kineticsTools inspects the alignment between the observed bases and the reference sequence -- in order for an IPD measurement to be included in the analysis, the PacBio read sequence must match the reference sequence for k around the cognate base. In the current module $k=1$. The IPD distribution at some locus be thought of as a mixture between the 'normal' incorporation process IPD, which is sensitive to the local sequence context and DNA modifications and a contaminating 'pause' process IPD which have a much longer duration (mean > 10x longer than normal), but happen rarely (~1% of IPDs). Note: Our current understanding is that pauses do not carry useful information about the methylation state of the DNA, however a more careful analysis may be warranted. Also note that modifications that drastically increase the IPD. Roughly 1% of observed IPDs are generated by pause events. Capping observed IPDs at the global 99th percentile is motivated by theory from robust hypothesis testing. Some sequence contexts may have naturally longer IPDs, to avoid capping too much data at those contexts, the cap threshold is adjusted per context as follows: `capThreshold = max(global99, 5*modelPrediction, percentile(ipdObservations, 75))`

Statistical Testing

We test the hypothesis that IPDs observed at a particular locus in the sample have a longer means than IPDs observed at the same locus in unmodified DNA. If we have generated a Whole Genome Amplified dataset, which removes DNA modifications, we use a case-control, two-sample t-test. This tool also provides a pre-calibrated 'synthetic control' model which predicts the unmodified IPD, given a 12 base sequence context. In the synthetic control case we use a one-sample t-test, with an adjustment to account for error in the synthetic control model.

Example Usage

Basic use with BAM input, GFF+HDF5 output:

```
ipdSummary aligned.bam --reference ref.fasta --identify m6A,m4C --gff basemods.gff --csv_h5 kinetics.h5
```

With cmp.h5 input, methyl fraction calculation and GFF+CSV output:

```
ipdSummary aligned.cmp.h5 --reference ref.fasta --identify m6A,m4C --methylFraction --gff basemods.gff --csv kinetics
```

Inputs

Aligned Reads

A standard PacBio alignment file - either AlignmentSet XML, BAM, or cmp.h5 - containing alignments and IPD information supplies the kinetic data used to perform modification detection. The standard cmp.h5 file of a SMRTportal jobs is `data/aligned_read.cmp.h5`.

Reference Sequence

The tool requires the reference sequence used to perform alignments. This can be either a FASTA file or a ReferenceSet XML.

Outputs

The modification detection tool provides results in a variety of formats suitable for in-depth statistical analysis, quick reference, and consumption by visualization tools such as PacBio SMRTView. Results are generally indexed by reference position and reference strand. In all cases the strand value refers to the strand carrying the modification in DNA sample. Remember that the kinetic effect of the modification is observed in read sequences aligning to the opposite strand. So reads aligning to the positive strand carry information about modification on the negative strand and vice versa, but in this toolkit we always report the strand containing the putative modification.

The following output options are available:

- `--gff FILENAME` : GFF format
- `--csv FILENAME` : comma-separated value format
- `--csv_h5 FILENAME` : compact binary equivalent of CSV in HDF5 format
- `--bigwig FILENAME` : BigWig file (mostly only useful for SMRTView)

If you are running base modification analysis through SMRT Link or a pbsmrtpipe pipeline, the GFF, HDF5, and BigWig outputs are automatically generated.

modifications.gff

The modifications.gff is compliant with the GFF Version 3 specification (<http://www.sequenceontology.org/gff3.shtml>). Each template position / strand pair whose p-value exceeds the pvalue threshold appears as a row. The template position is 1-based, per the GFF spec. The strand column refers to the strand carrying the detected modification, which is the opposite strand from those used to detect the modification. The GFF confidence column is a Phred-transformed pvalue of detection.

Note on genome browser compatibility

The modifications.gff file will not work directly with most genome browsers. You will likely need to make a copy of the GFF file and convert the `_seqid_` columns from the generic 'ref0000x' names generated by PacBio, to the FASTA headers present in the original reference FASTA file. The mapping table is written in the header of the modifications.gff file in `#sequence-header` tags. This issue will be resolved in the 1.4 release of kineticsTools.

The auxiliary data column of the GFF file contains other statistics which may be useful downstream analysis or filtering. In particular the coverage level of the reads used to make the call, and +/- 20bp sequence context surrounding the site.

modifications.csv

The modifications.csv file contains one row for each (reference position, strand) pair that appeared in the dataset with coverage at least `x`. `x` defaults to 3, but is configurable with `'--minCoverage'` flag to `ipdSummary.py`. The reference position index is 1-based for compatibility with the gff file the R environment. Note that this output type scales poorly and is not recommended for large genomes; the HDF5 output should perform much better in these cases. We have preserved the CSV option to support legacy applications but this is no longer produced by the pipelines in SMRT Link/pbsmrtpipe.

modifications.h5

The HDF5 output largely mirrors the CSV output in content, but is structured slightly differently. Each contig in the reference has its own group in the file, keyed by FASTA ID. For each group, the columns in the CSV file are represented as arrays:

```
modifications.h5
--> refName
--> tpl
--> strand
--> base
--> score
--> tMean
--> tErr
--> modelPrediction
```

```
--> ipdRatio
--> coverage
```

For example, the following code to iterate over the CSV file:

```
import csv
with open("modifications.csv") as f:
    for rec in csv.reader(f):
        process_record(rec)
```

translates approximately to this code for reading the HDF5:

```
import h5py
COLUMNS="refName,tpl,strand,base,score,tMean,tErr,modelPrediction,ipdRatio,coverage".split(",")
with h5py.File(file_name) as f:
    for ctg_id in sorted(f.keys()):
        values = f[ctg_id]
        for i in range(len(values["tpl"])):
            rec = [ctg_id] + [fmt(values[k][i]) for k in COLUMNS[1:]]
            process_record(rec)
```

Note that the exact columns present in both files may vary depending on how kineticsTools was run; however, the example above is valid for the results of the pbsmrtpipe base modification analysis pipelines.

Output columns

in-silico control mode

Column	Description
refId	reference sequence ID of this observation
tpl	1-based template position
strand	native sample strand where kinetics were generated. '0' is the strand of the original FASTA, '1' is opposite strand from FASTA
base	the cognate base at this position in the reference
score	Phred-transformed pvalue that a kinetic deviation exists at this position
tMean	capped mean of normalized IPDs observed at this position
tErr	capped standard error of normalized IPDs observed at this position (standard deviation / sqrt(coverage))
modelPrediction	normalized mean IPD predicted by the synthetic control model for this sequence context
ipdRatio	tMean / modelPrediction
coverage	count of valid IPDs at this position (see Filtering section for details)
frac	estimate of the fraction of molecules that carry the modification
fracLow	2.5% confidence bound of frac estimate
fracUpp	97.5% confidence bound of frac estimate

case-control mode

Column	Description
refId	reference sequence ID of this observation

Column	Description
tpl	1-based template position
strand	native sample strand where kinetics were generated. '0' is the strand of the original FASTA, '1' is opposite strand from FASTA
base	the cognate base at this position in the reference
score	Phred-transformed pvalue that a kinetic deviation exists at this position
caseMean	mean of normalized case IPDs observed at this position
controlMean	mean of normalized control IPDs observed at this position
caseStd	standard deviation of case IPDs observed at this position
controlStd	standard deviation of control IPDs observed at this position
ipdRatio	tMean / modelPrediction
testStatistic	t-test statistic
coverage	mean of case and control coverage
controlCoverage	count of valid control IPDs at this position (see Filtering section for details)
caseCoverage	count of valid case IPDs at this position (see Filtering section for details)

