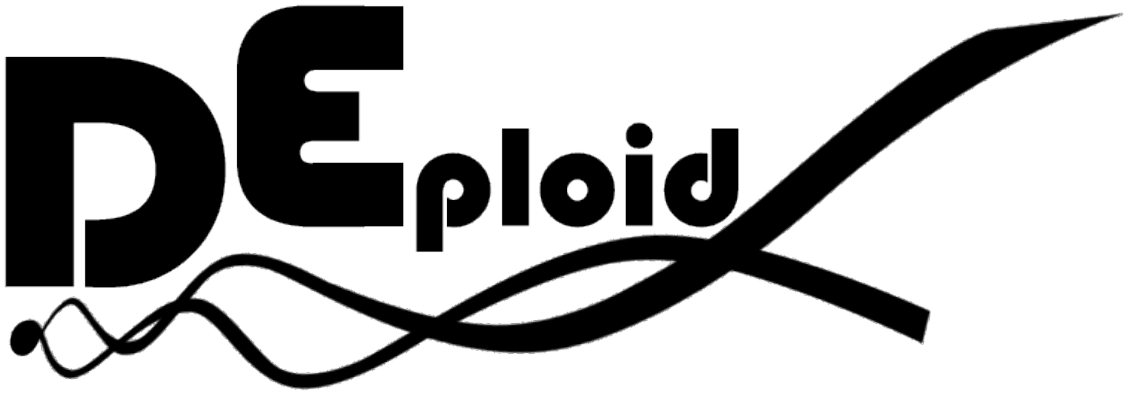


DEploid Documentation

Release v0.6-beta

Sha (Joe) Zhu

Nov 18, 2019

Contents

1 Synopsis **2**
- 1.1 Example: . . . 2

2 Description **3**

3 Installation **5**
- 3.1 Stable Release . . . 5
- 3.2 Development Version From GitHub . . . 5

4 How it works? **7**
- 4.1 Program parameters and options . . . 7
- 4.2 Example of data exploration . . . 9

5 Making sense of the output **11**
- 5.1 Output files . . . 11
- 5.2 Example of output interpretation . . . 12

6 Pf3k workflow **16**

7 Frequently asked questions **17**
- 7.1 Data filtering . . . 17
- 7.2 Over-fitting . . . 21
- 7.3 Benchmark . . . 24

8 Reporting Bugs **25**

9 Citing DEploid **26**

Bibliography **28**

A software that deconvolutes mixed genomes with unknown proportions.

CHAPTER 1

Synopsis

dEploid [-vcf *file*] [-plaf *file*] [-noPanel] . . . [-exclude *file*] [-vcfOut] [-o *string*]

dEploid [-vcf *file*] [-plaf *file*] [-panel *file*] . . . [-exclude *file*] [-vcfOut] [-o *string*]

1.1 Example:

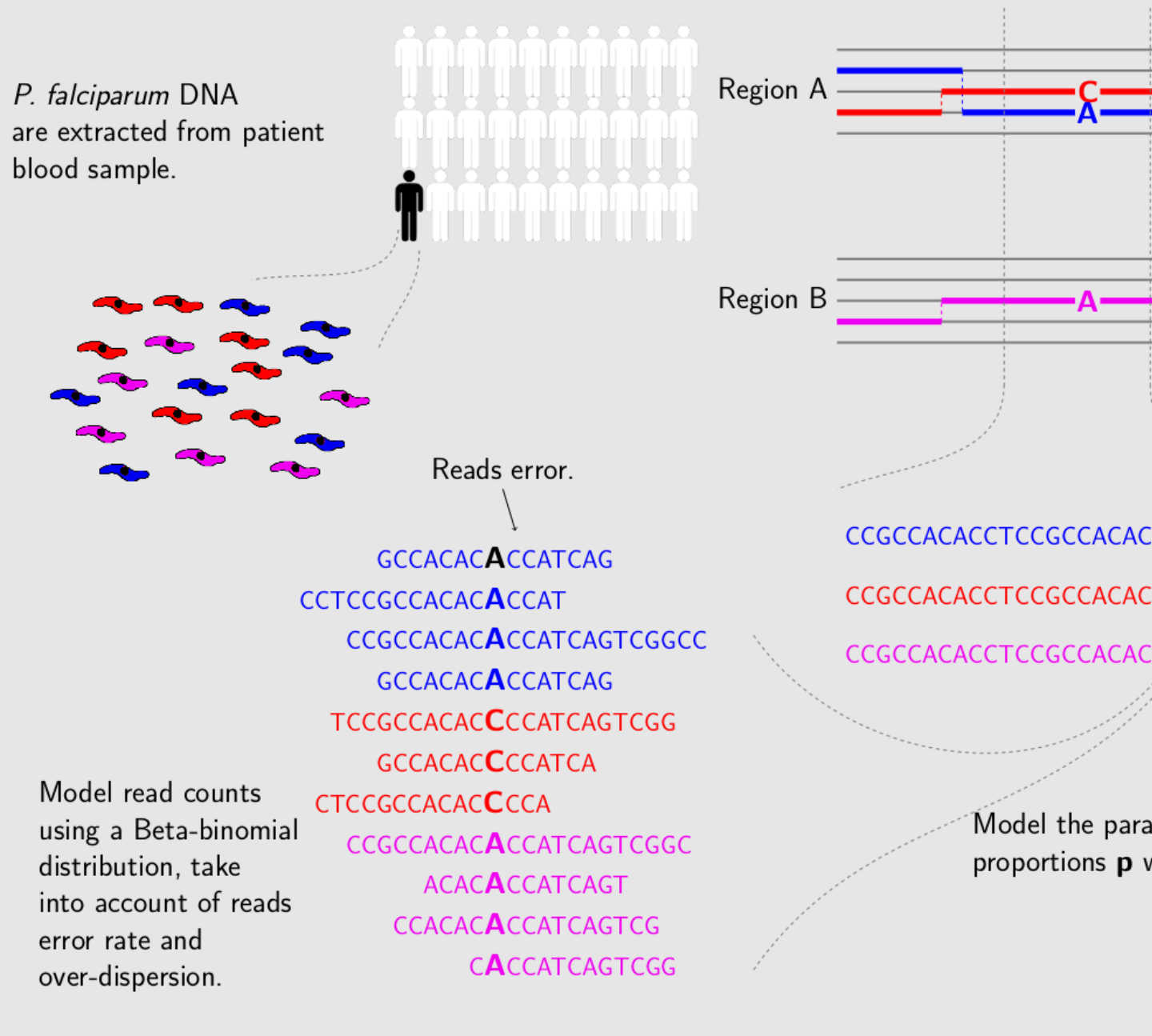
```
$ ./dEploid -vcf data/exampleData/PG0390-C.eg.vcf.gz \
-plaf data/exampleData/labStrains.eg.PLAF.txt \
-exclude data/testData/labStrains.test.exclude.txt \
-noPanel \
-o test_run \
-vcfOut -z
```

CHAPTER 2

Description

`dEploid` is designed for deconvoluting mixed genomes with unknown proportions. Traditional 'phasing' programs are limited to diploid organisms. Our method modifies Li and Stephen's [Li2003] algorithm with Markov chain Monte Carlo (MCMC) approaches, and builds a generic framework that allows haplotype searches in a multiple infection setting.

`dEploid` is primarily developed as part of the Pf3k project, from which this documentation will take examples from for demonstration. The Pf3k project is a global collaboration using the latest sequencing technologies to provide a high-resolution view of natural variation in the malaria parasite *Plasmodium falciparum*. Parasite DNA are extracted from patient blood sample, which often contains more than one parasite strain, with unknown proportions. `DEploid` is used for deconvoluting mixed haplotypes, and reporting the mixture proportions from each sample.


P. falciparum DNA are extracted from patient blood sample.
Region A
C
A
Region B
A
Reads error.
GCCACACACCATCAG
CCTCCGCCACACACCAT
CCGCCACACACCATCAGTCGGCC
GCCACACACCATCAG
TCCGCCACACCCATCAGTCGG
GCCACACCCCATCA
CTCCGCCACACCCCA
CCGCCACACACCATCAGTCGGC
ACACACCATCAGT
CCACACACCATCAGTCG
CACCATCAGTCGG
CCGCCACACCTCCGCCACAC
CCGCCACACCTCCGCCACAC
CCGCCACACCTCCGCCACAC
Model read counts using a Beta-binomial distribution, take into account of reads error rate and over-dispersion.
Model the para
proportions **p** v

CHAPTER 3

Installation

dEploid is written in C++.

3.1 Stable Release

The latest version of dEploid is available at github.

3.2 Development Version From GitHub

You can also install dEploid directly from the git repository (tar, zip). Here, you will need autoconf, check whether this is already installed by running:

```
$ which autoconf
```

On Debian/Ubuntu based systems:

```
$ apt-get install build-essential autoconf autoconf-archive libcppunit-dev
```

On Mac OS:

```
$ port install automake autoconf autoconf-archive cppunit
```

Afterwards you can clone the code from the github repository,

```
$ git clone git@github.com:mcveanlab/DEploid.git
$ cd DEploid
```

and build the binary using

```
$ ./bootstrap
$ make
```

or install with `make install`.

CHAPTER 4

How it works?

4.1 Program parameters and options

4.1.1 Mostly used

-vcf [*file*] File path of the isolate vcf. Assume all variants are `PASS` in the `QUAL` column, the VCF file also reqires the `AD` field.

Note: In the current implementation, `DEploid` only take the first sample in the VCF file. `DEploid` DO NOT handle multi-allelic variants, nor indels. The `FILTER` column will not be used.

-plaf [*file*] File path of population level allele frequencies (tab-delimited plain text file), for example

CHROM	POS	PLAF
Pf3D7_01_v3	93157	0.0190612159917058
Pf3D7_01_v3	94422	0.135502358766423
Pf3D7_01_v3	94459	0.156294363760064
Pf3D7_01_v3	94487	0.143439298925837

-panel [*file*] File path of the reference panel (tab-delimited plain text file), for example

CHROM	POS	3D7	Dd2	Hb3	7G8
Pf3D7_01_v3	93157	0	0	0	1
Pf3D7_01_v3	94422	0	0	0	1
Pf3D7_01_v3	94459	0	0	0	1
Pf3D7_01_v3	94487	0	0	0	1

-noPanel Use population level allele frequency as prior.

Warning: Flags **-panel** and **-noPanel** should not be used together.

-exclude [file] File path of sites to be excluded (tab-delimited plain text file).

-o [string] Specify the file name prefix of the output.

-k [int] Number of strain (default value 5).

-seed [int] Random seed.

-nSample [int] Number of MCMC samples (default value 800).

-rate [int] MCMC sample rate (default value 5).

-burn [float] MCMC burn rate (default value 0.5).

-ibd Use IBD segment to infer the proportion, then infer the haplotype (see Pf3k work-flow for more details).

-painting [*file*] Paint the posterior probability of the given haplotypes.

-inbreeding Calculate the inbreeding probabilities.

-initialP [float . . .] Initialize proportions.

-ibdPainting IBD painting, compute posterior probabilities of IBD configurations of given strain proportions. This option must be used with flags *-initialP*.

-h , -help Help.

-v , -version DEploid version.

-vcfOut Save final halpotypes into a VCF file.

4.1.2 You may also try

-ref [file] File path of reference allele count (tab-delimited plain text file).

Note: In early `dEploid` versions (prior to *v0.2-release*), allele counts extracted from the vcf file are placed in two files, and parsed by flags **-ref [file]** and **-alt [file]**. Tab-delimited plain text for input. First and second columns record chromosome and position labels respectively. Third columns records the reference allele count or alternative allele count. For example,

Table 1: Reference allele count

CHROM	POS	PG0390.C
Pf3D7_01_v3	93157	85
Pf3D7_01_v3	94422	77
Pf3D7_01_v3	94459	90
Pf3D7_01_v3	94487	79

-alt [file] File path of alternative allele count (tab-delimited plain text file).

Table 2: Alternative allele count

CHROM	POS	PG0390.C
Pf3D7_01_v3	93157	0
Pf3D7_01_v3	94422	0
Pf3D7_01_v3	94459	0
Pf3D7_01_v3	94487	0

Warning: Flags **-ref** and **-alt** should not be used with **-vcf**.

-forbidUpdateProp Forbid MCMC moves to update proportions.

-forbidUpdateSingle Forbid MCMC moves to update single haplotype.

-forbidUpdatePair Forbid MCMC moves to update pair haplotypes.

-exportPostProb Save the posterior probabilities of the final iteration of all strains.

-miss [float] Miss copying probability.

-recomb [float] Constant recombination probability.

-p [int] Output precision (default value 8).

-c [float] Specify scaling parameter c, which reflects how much data is available (default value 100.0).

-G [float] Specify scaling parameter for genetic map (default value of 20.0).

-sigma [float] Specify the variance parameter for proportion estimation (default value of 5.0).

-ibdSigma [flat] Specify the variance parameter for proportion estimation when IBD method is used (default value of 20.0).

-initialHap [file] Specify initial haplotypes of deconvolution.

4.1.3 R utilities

Flags **-vcf**, **-plaf**, **-ref**, **-alt**, **-exclude**, **-o** usage are the same as `DEploid`. Additionally, we have the following flags:

-dEprefix [string] Prefix of `DEploid` output.

-inbreeding Painting haplotype inbreeding posterior probabilities.

-ADFieldIndex The index of `AD` field (2 by default).

-filter.threshold [float] Filtering threshold (0.995 by default).

-filter.window [int] Filtering window (10 by default).

-pdf Produce figures in pdf rather than png.

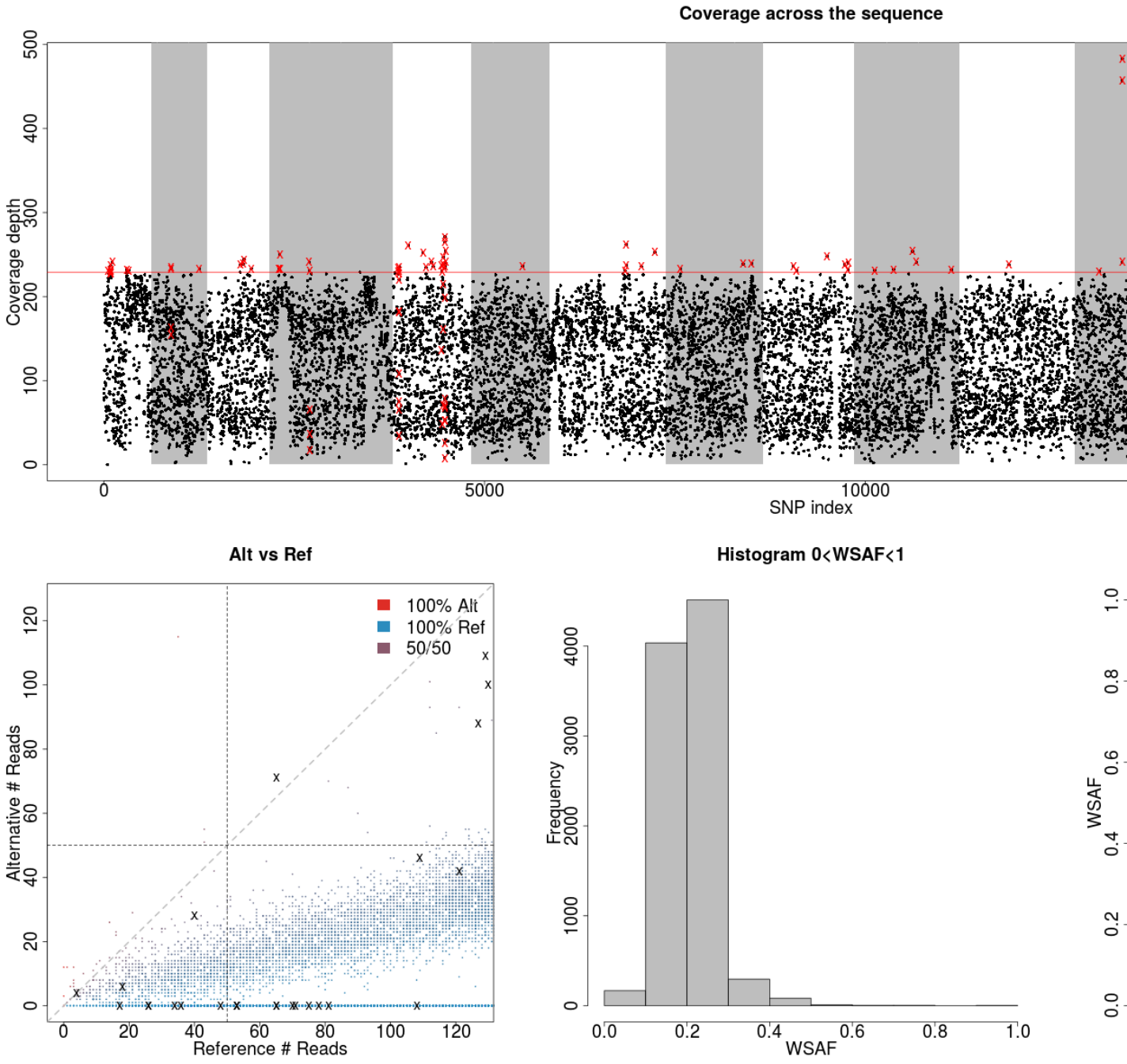
-ibd Produce figures for IBD process.

-ring Produce circular genome plots for WSAF and haplotype posterior painting probabilities.

4.2 Example of data exploration

Use our data exploration tools to investigate the data.

```
$ utilities/dataExplore.r -vcf data/exampleData/PG0390-C.eg.vcf.gz \
 -plaf data/exampleData/labStrains.eg.PLAF.txt \
 -o PG0390-C
```



- Figure on the top plot total allele counts across all markers. We use the threshold (red line) to identify markers with extremely high allele counts. Red crosses indicate markers that are filtered out.
- Figure on the left plots the alternative allele count against the reference allele count. As *P. falciparum* genomes are haploid, in clonal samples, one woule expect to see either alternative or reference allele at any sites. Heterozygous sites are indications of mixed infection.
- Figure in the middle is the histogram of the allele frequency within sample. Note that we exclude markers with WSAF strictly equal to 0s and 1s in the histogram.
- Figure on the right show allele frequency within sample, compare against the population average.

CHAPTER 5

Making sense of the output

5.1 Output files

`dEploid` outputs text files with user-specified prefix with flag **-o**.

***prefix*.log**

Log file records `dEploid` version, input file paths, parameter used and proportion estimates at the final iteration.

***prefix*.llk**

Log likelihood of the MCMC chain.

***prefix*.prop**

MCMC updates of the proportion estimates.

***prefix*.hap**

Haplotypes at the final iteration in plain text file.

***prefix*.vcf**

When flag `-vcfOut` is turned on, haplotypes are saved at the final iteration in VCF format.

***prefix*.single[i]**

When flag `-exportPostProb` is turned on, posterior probabilities of the final iteration of strain [i].

5.1.1 DEploid-IBD

When "flag" `-ibd` is used. 'DEploid' executes first learns the number of strain and their proportions with an identity by descent model ('DEploid-IBD'). Then it fixes the number of strains and proportions and train the haplotypes, and train the haplotypes using the original DEploid algorithm ('DEploid-classic'). The staged output are labelled with ".ibd" and ".classic" respectively, and followed by the prefix.

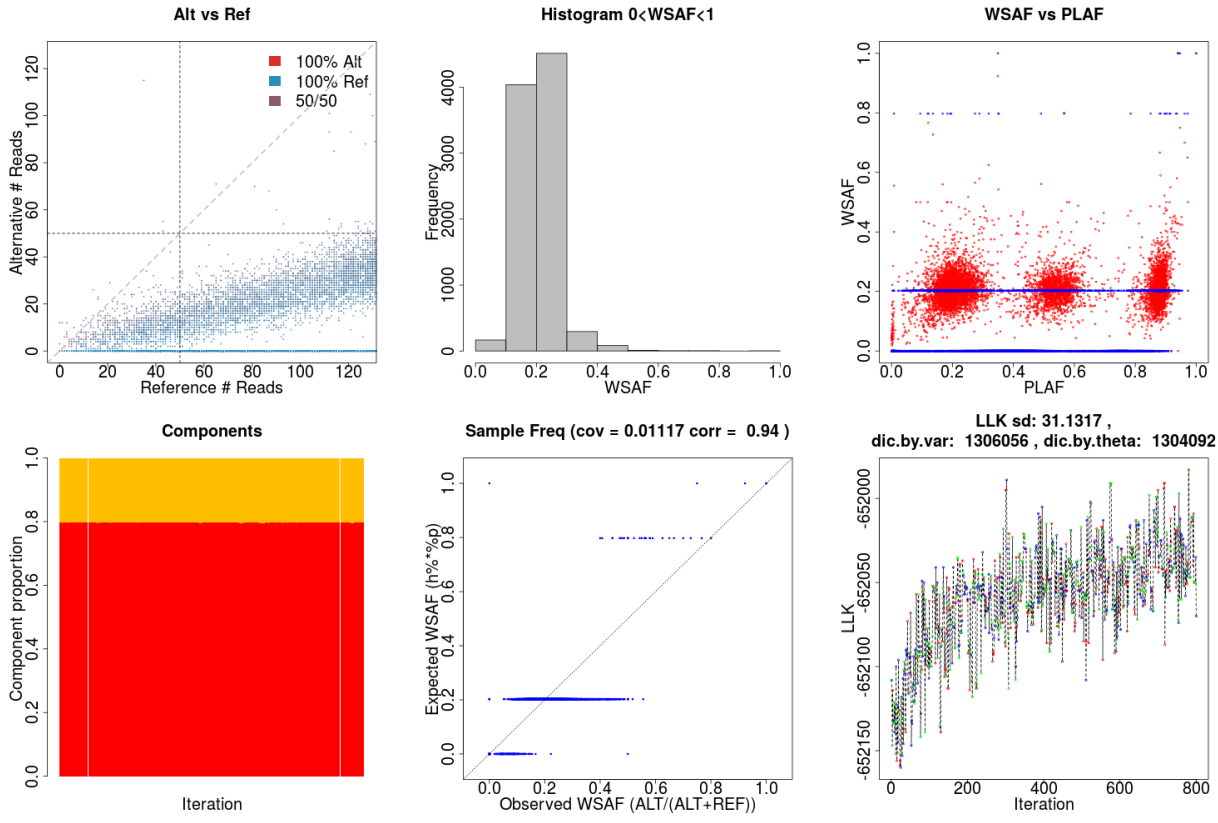
5.1.2 DEploid-BEST

When "flag" `-best` is used. 'DEploid-BEST' executes the deconvolution algorithms in an optimised sequence to best report the number of strains, proportions and haplotypes. The program ('DEploid-Lasso') learns the number of strain with optimised reference panel; ".chooseK" is appended to the prefix for these output (NOTE: likelihood is not tracked in this case). It ('DEploid-IBD') then fixes the number of strains and tune the strain proportions with an identity by descent model; ".ibd" is appended to the prefix for these output. Finally, the program ('DEploid-Lasso') fixes the number of strains and proportions, and uses the optimised reference panel again to train and report the haplotypes; ".final" is appended to the prefix for these output. When `-vcfOut` is applied, this will only be the final haplotypes.

5.2 Example of output interpretation

5.2.1 Example 1. Standard deconvolution output

```
$ ./dEploid -vcf data/exampleData/PG0390-C.eg.vcf.gz \
-plaf data/exampleData/labStrains.eg.PLAF.txt \
-noPanel -o PG0390-CNopanel -seed 1
$ utilities/interpretDEploid.r -vcf data/exampleData/PG0390-C.eg.vcf.gz \
-plaf data/exampleData/labStrains.eg.PLAF.txt \
-dEprefix PG0390-CNopanel \
-o PG0390-CNopanel -ring
```

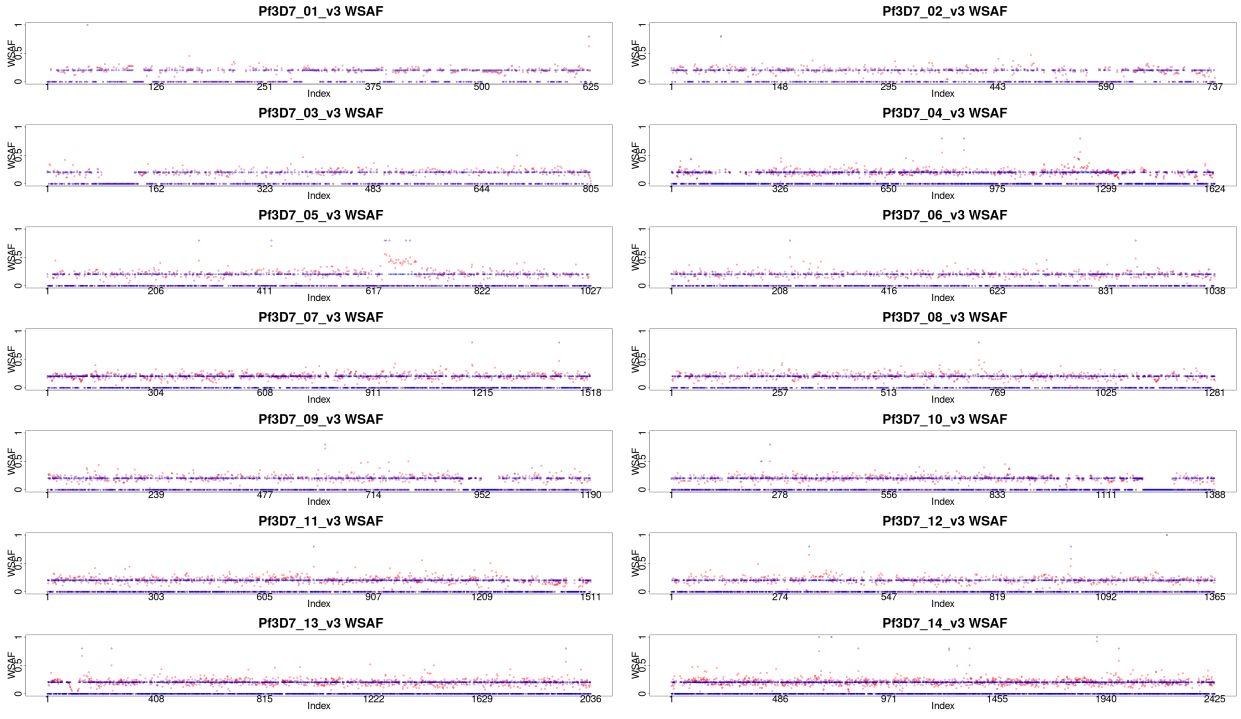


interpretDEploidF

The top three figures are the same as figures show in :ref:`data example <sec-eg>`, with a small addition of inferred WSAF marked in blue, in the top right figure.

- The bottom left figure show the relative proportion change history of the MCMC chain.

- The middle figure show the correlation between the expected and observed allele frequency in sample.
- The right figure shows changes in MCMC likelihood .



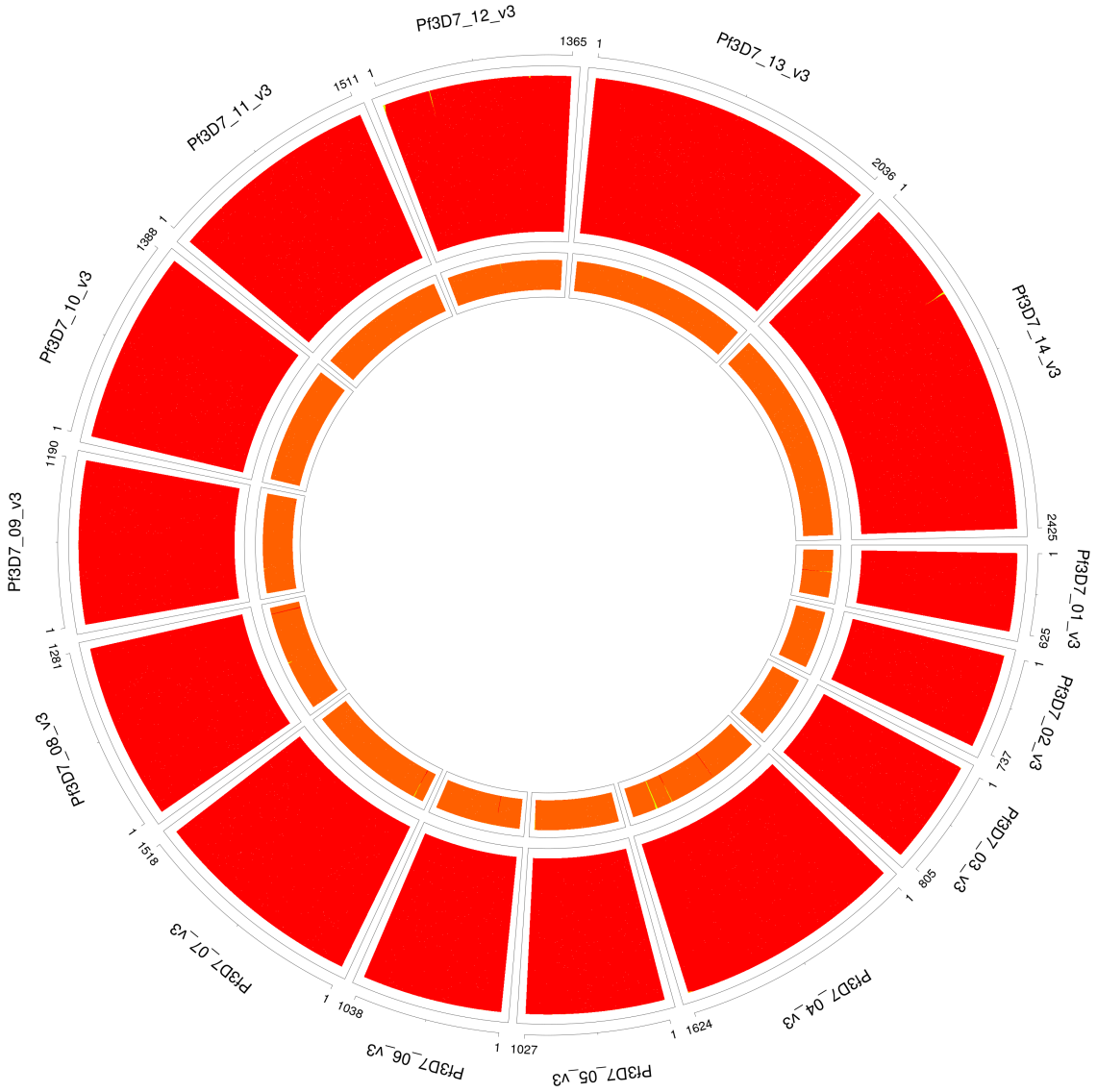
interpretDEploidF

This panel figure shows all allele frequencies within sample across all 14 chromosomes. Expected and observed WSAF are marked in blue and red respectively.

5.2.2 Example 2. Haplotype painting from a given panel

`dEploid` can take its output haplotypes, and calculate the posterior probability of each deconvoluted strain with the reference panel. In this example, the reference panel includes four lab strains: 3D7 (red), Dd2 (dark orange), HB3 (orange) and 7G8 (yellow).

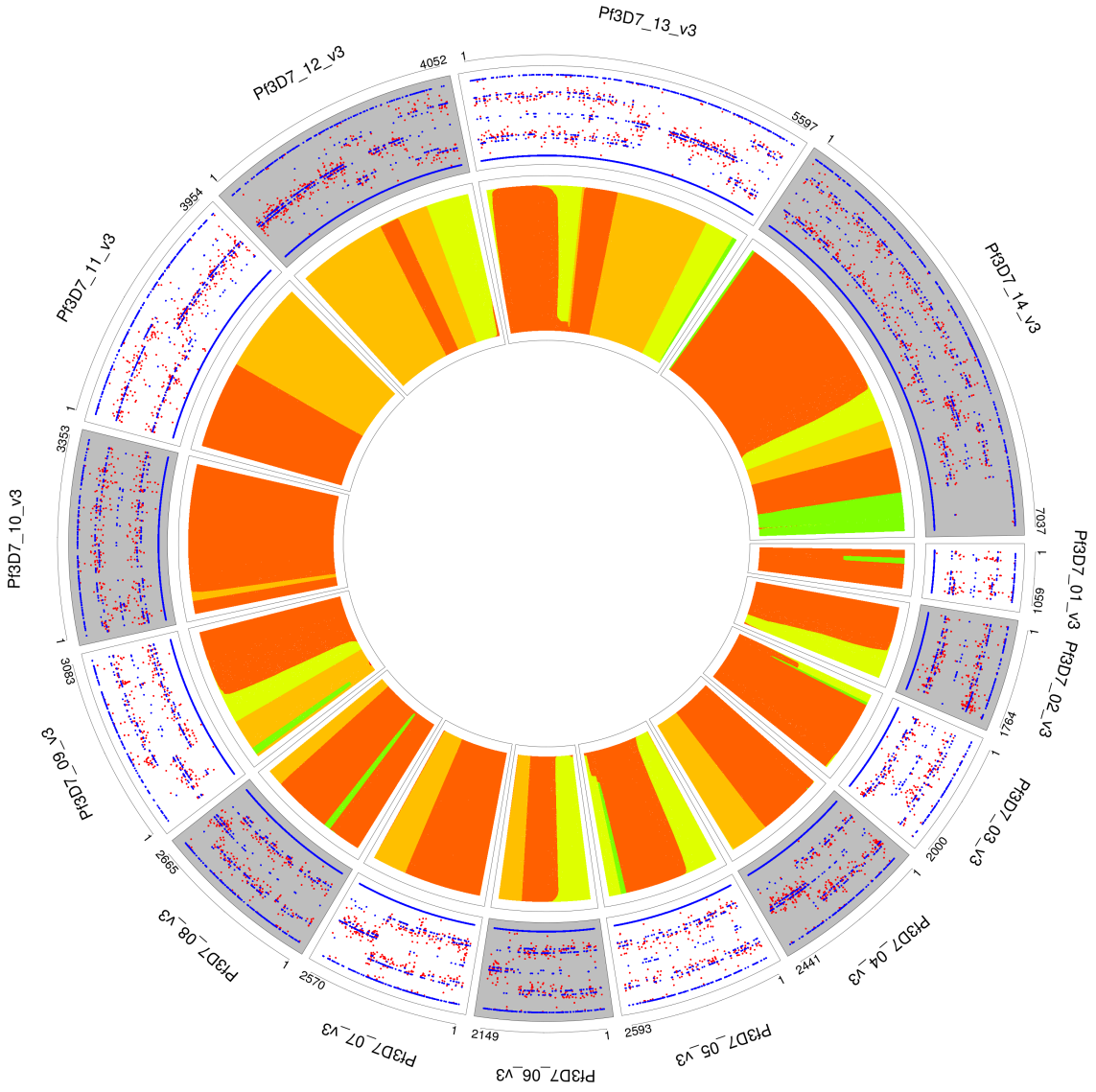
```
$ ./dEploid -vcf data/exampleData/PG0390-C.eg.vcf.gz \
-plaf data/exampleData/labStrains.eg.PLAF.txt \
-panel data/exampleData/labStrains.eg.panel.txt \
-o PG0390-CPanel -seed 1 -k 3
$ ./dEploid -vcf data/exampleData/PG0390-C.eg.vcf.gz \
-plaf data/exampleData/labStrains.eg.PLAF.txt \
-panel data/exampleData/labStrains.eg.panel.txt \
-o PG0390-CPanel \
-painting PG0390-CPanel.hap \
-initialP 0.8 0 0.2 -k 3
$ utilities/interpretDEploid.r -vcf data/exampleData/PG0390-C.eg.vcf.gz \
-plaf data/exampleData/labStrains.eg.PLAF.txt \
-dEprefix PG0390-CPanel \
-o PG0390-CPanel -ring
```



PG0390fwdBwdR

5.2.3 Example 3. Deconvolution followed by IBD painting

In addition to lab mixed samples, here we show example of `dEploid` deconvolute field sample PD0577-C.



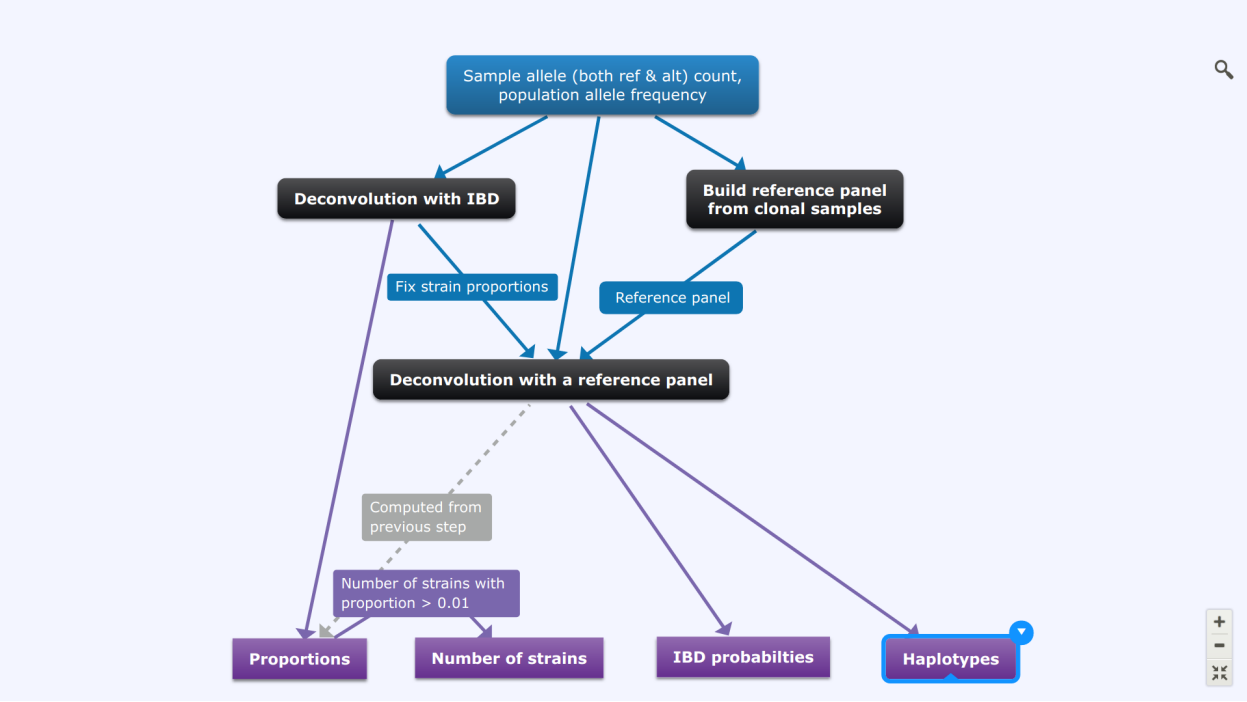
PD0577inbreeding

CHAPTER 6

Pf3k workflow

Our main work flow consist with three steps:

1. Use `dEploid` on clonal samples, and build a reference panel.
2. Use the IBD method to infer the proportions without a reference panel.
3. Tune the haplotype with the given reference panel with fixed strain proportions



workflow

Black boxes indicate the key deconvolution steps when our program DEploid is used. Boxes in blue and purple represent the input and output respectively at each step. Steps **Deconvolution with IBD** and **Deconvolution with a reference panel** can be combined by using the flag `-ibd`.

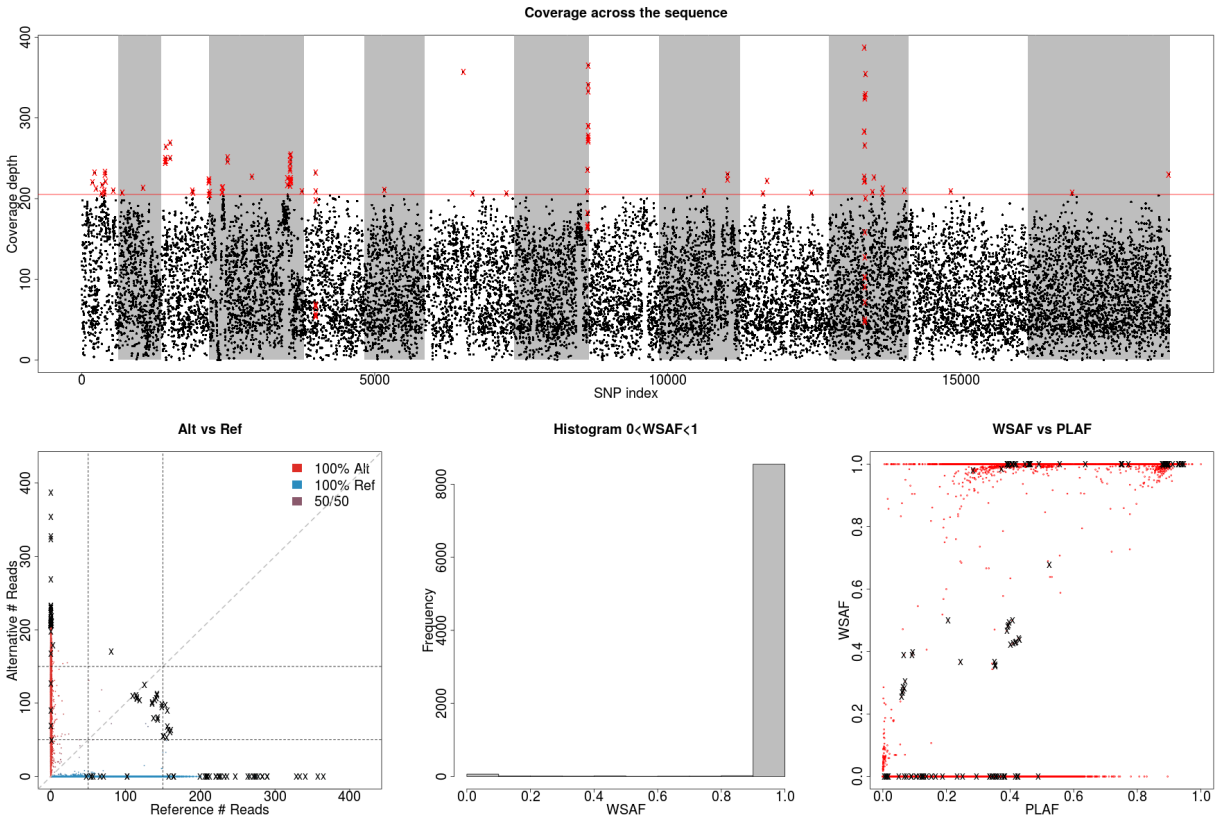
CHAPTER 7

Frequently asked questions

7.1 Data filtering

Data filtering is an important step for deconvolution.

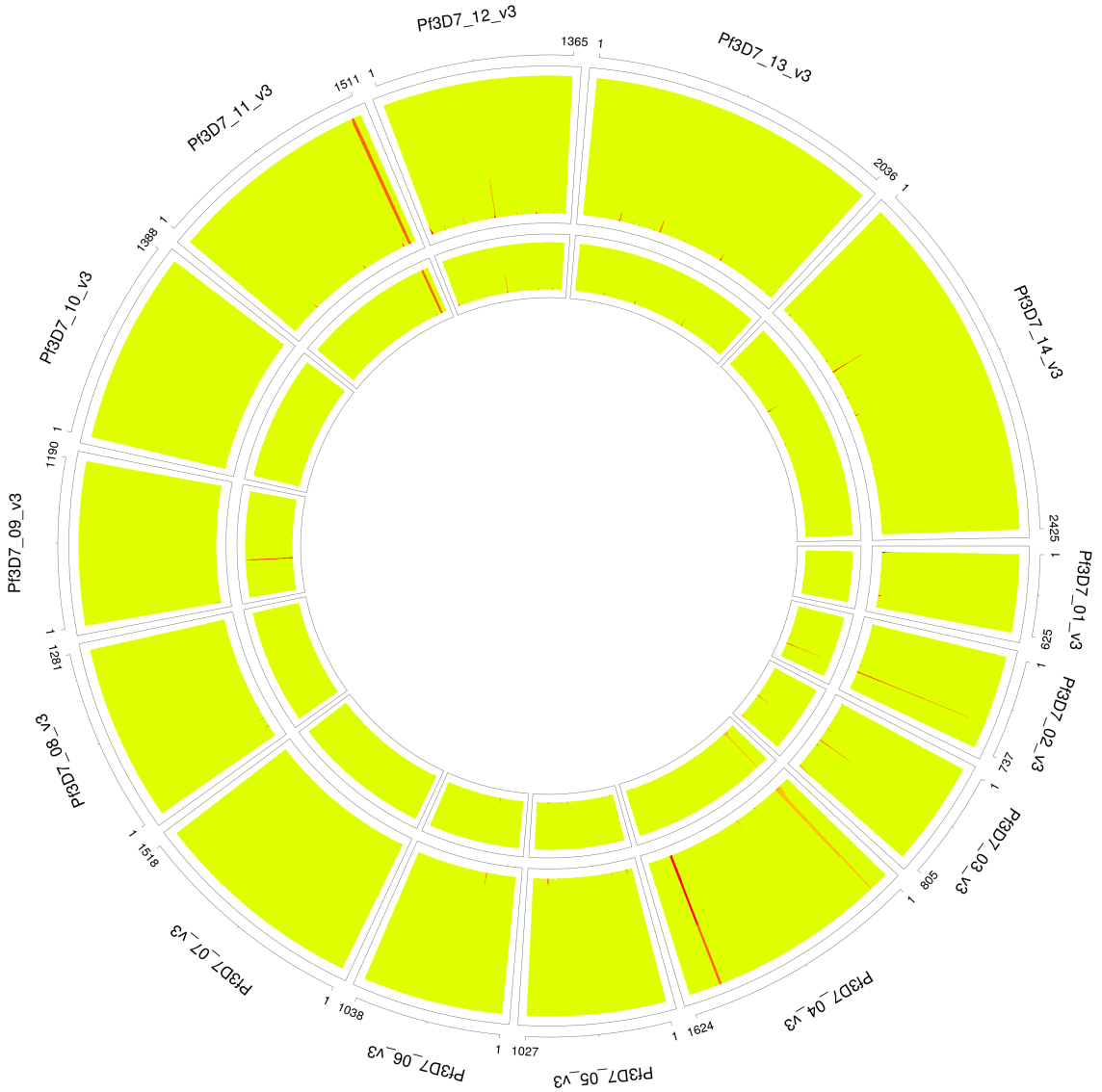
```
utilities/dataExplore.r -vcf data/exampleData/PG0415-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -o PG0415-C
```



PG0415_data

We observe a small number of heterozygous sites with high coverage (marked as crosses above), which can potentially mislead our model to over-fit the data with additional strains.

```
./dEploid -vf data/exampleData/PG0415-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -noPanel -o PG0415-CNopanel -seed 2

initialProp=$( cat PG0415-CNopanel.prop | tail -1 | sed -e "s/\t/ /g" )
./dEploid -vf data/exampleData/PG0415-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -panel data/exampleData/labStrains.eg.panel.txt \
    -o PG0415-CNopanel \
    -initialP ${initialProp} \
    -painting PG0415-CNopanel.hap

utilities/interpretDEploid.r -vf data/exampleData/PG0415-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -dEprefix PG0415-CNopanel \
    -o PG0415-CNopanel \
    -ring
```



PG0415_noFilter

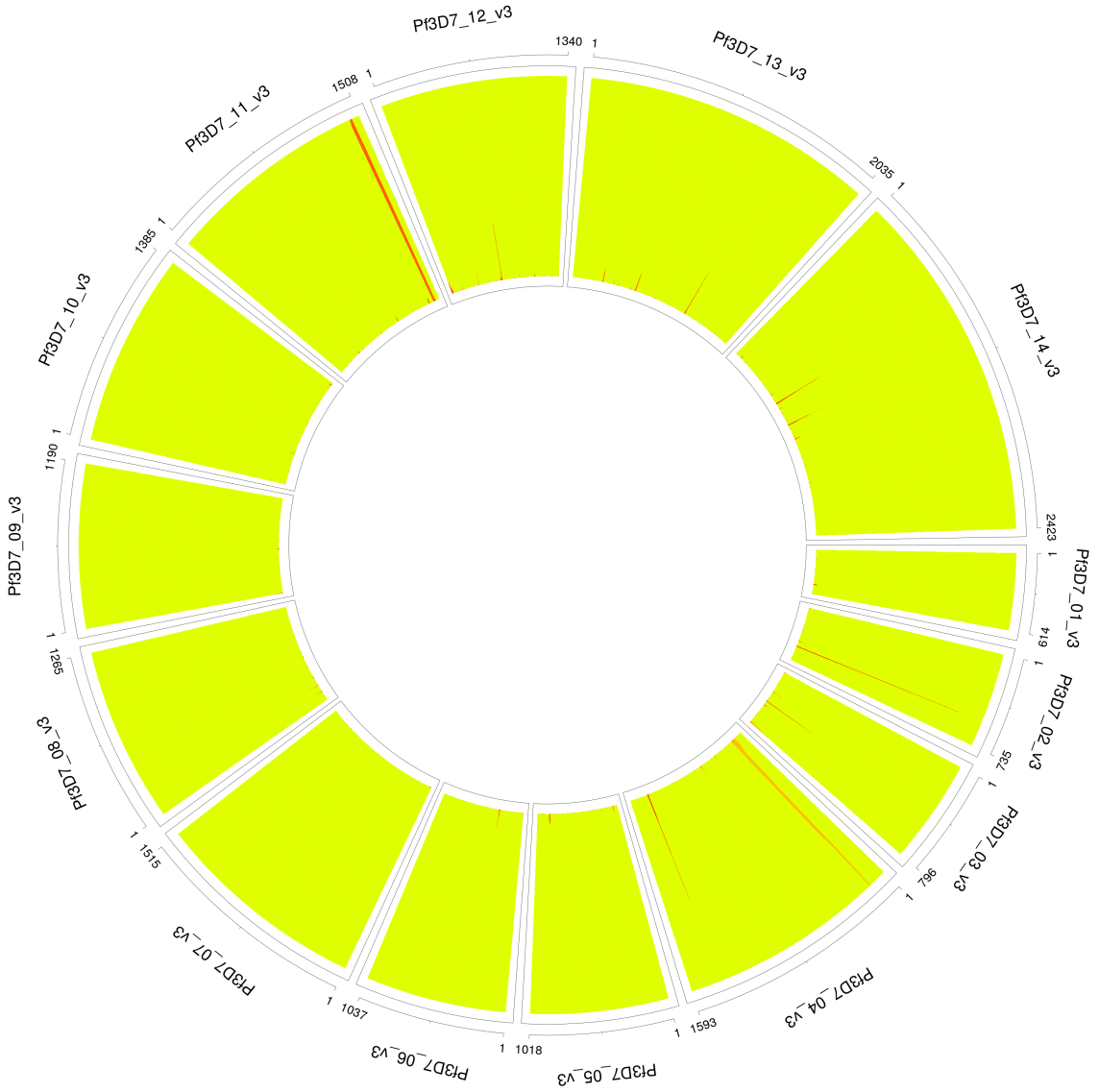
The data exploration utility `utilities/dataExplore.r` identifies a list of potential outliers. After filtering, we correctly identify the number of strains and proportion.

```
./dEploid -vcf data/exampleData/PG0415-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -noPanel -o PG0415-CNopanel.filtered -seed 2 \
    -exclude PG0415-CPotentialOutliers.txt

initialProp=$( cat PG0415-CNopanel.filtered.prop | tail -1 | sed -e "s/\t/ /g" )
./dEploid -vcf data/exampleData/PG0415-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -panel data/exampleData/labStrains.eg.panel.txt \
    -exclude PG0415-CPotentialOutliers.txt \
    -o PG0415-CNopanel.filtered \
```

(continues on next page)

(continued from previous page)

```
    -initialP ${initialProp} \
    -painting PG0415-CNopanel.filtered.hap

utilities/interpretDEploid.r -vcf data/exampleData/PG0415-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -dEprefix PG0415-CNopanel.filtered \
    -o PG0415-CNopanel.filtered \
    -exclude PG0415-CPotentialOutliers.txt \
    -ring
```

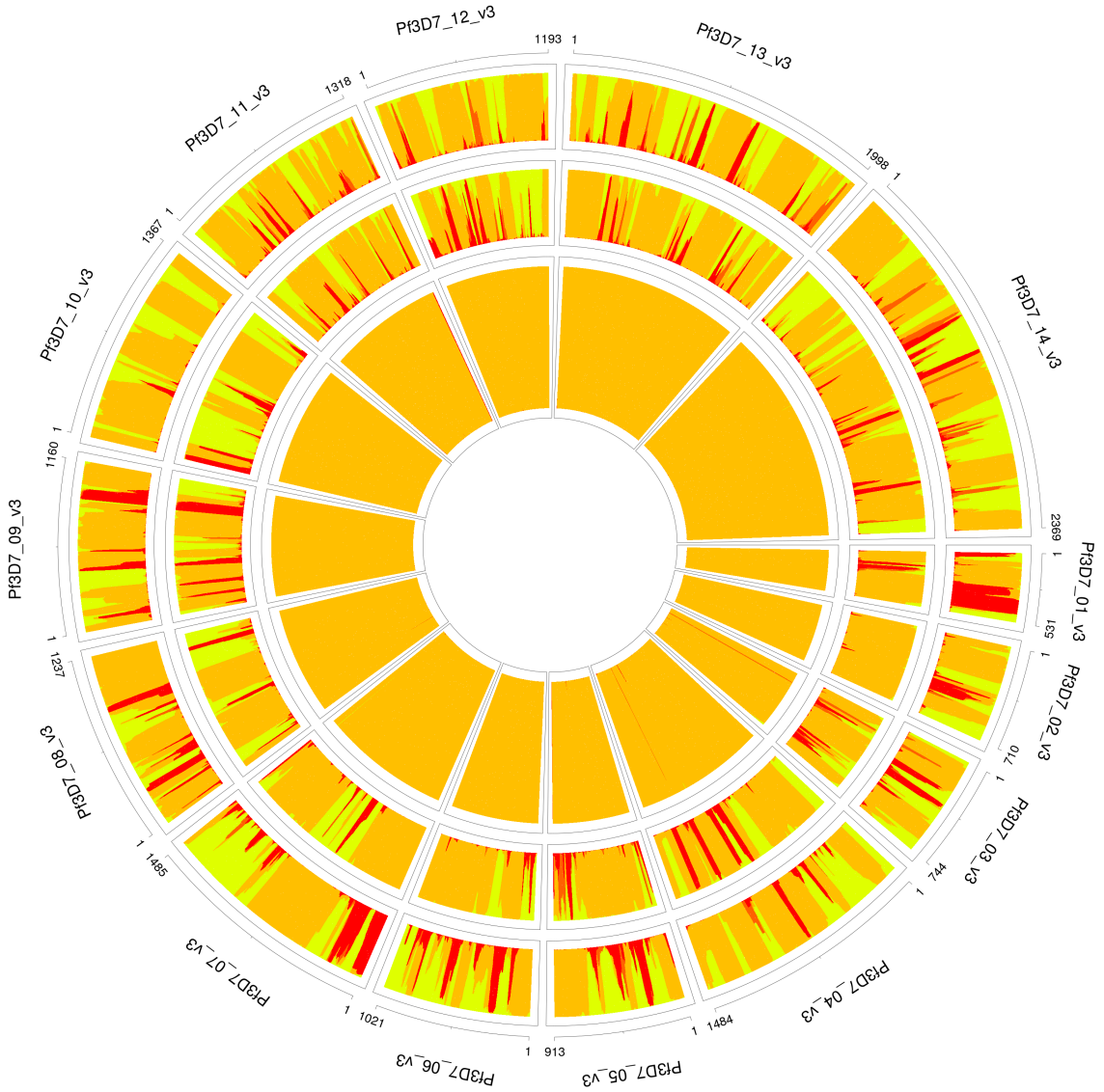


PG0415_filtered

7.2 Over-fitting

For extremely unbalanced samples, DEploid tends to over-fit the minor strain with an additional component. We recommend adjusting the value of sigma for the prior to improve inference. In this example PG0400-C is a mixture of lab strains 7G8 and HB3 with mixing proportions of 95/5%. The parameter sigma takes value of 5 by default, which over fits the minor strain (see *example 1*), and with proportions 0.0276862, 0.945509 and 0.0267463. *Example 1* paints the deconvolved strains (proportions in increasing order towards the centre) to the reference panel. We resolve the over-fitting issue by rerun this example, and set sigma with value of 10, it correctly infer the proportions as 0.0313755 and 0.968599 (see *example 2*). Note that the radius are not in scale with strain proportions.

```
./dEploid -vcf data/exampleData/PG0400-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -panel  data/exampleData/labStrains.eg.panel.txt \
    -o PG0400-Csigma5 -seed 2 -sigma 5 \
    -exclude exclude.txt

initialProp=$( cat PG0400-Csigma5.prop | tail -1 | sed -e "s/\t/ /g" )
./dEploid -vcf data/exampleData/PG0400-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -panel data/exampleData/labStrains.eg.panel.txt \
    -exclude exclude.txt \
    -o PG0400-Csigma5 \
    -initialP ${initialProp} \
    -painting PG0400-Csigma5.hap

utilities/interpretDEploid.r -vcf data/exampleData/PG0400-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -dEprefix PG0400-Csigma5 \
    -o PG0400-Csigma5 \
    -exclude exclude.txt \
    -reverseRing -transformP
```



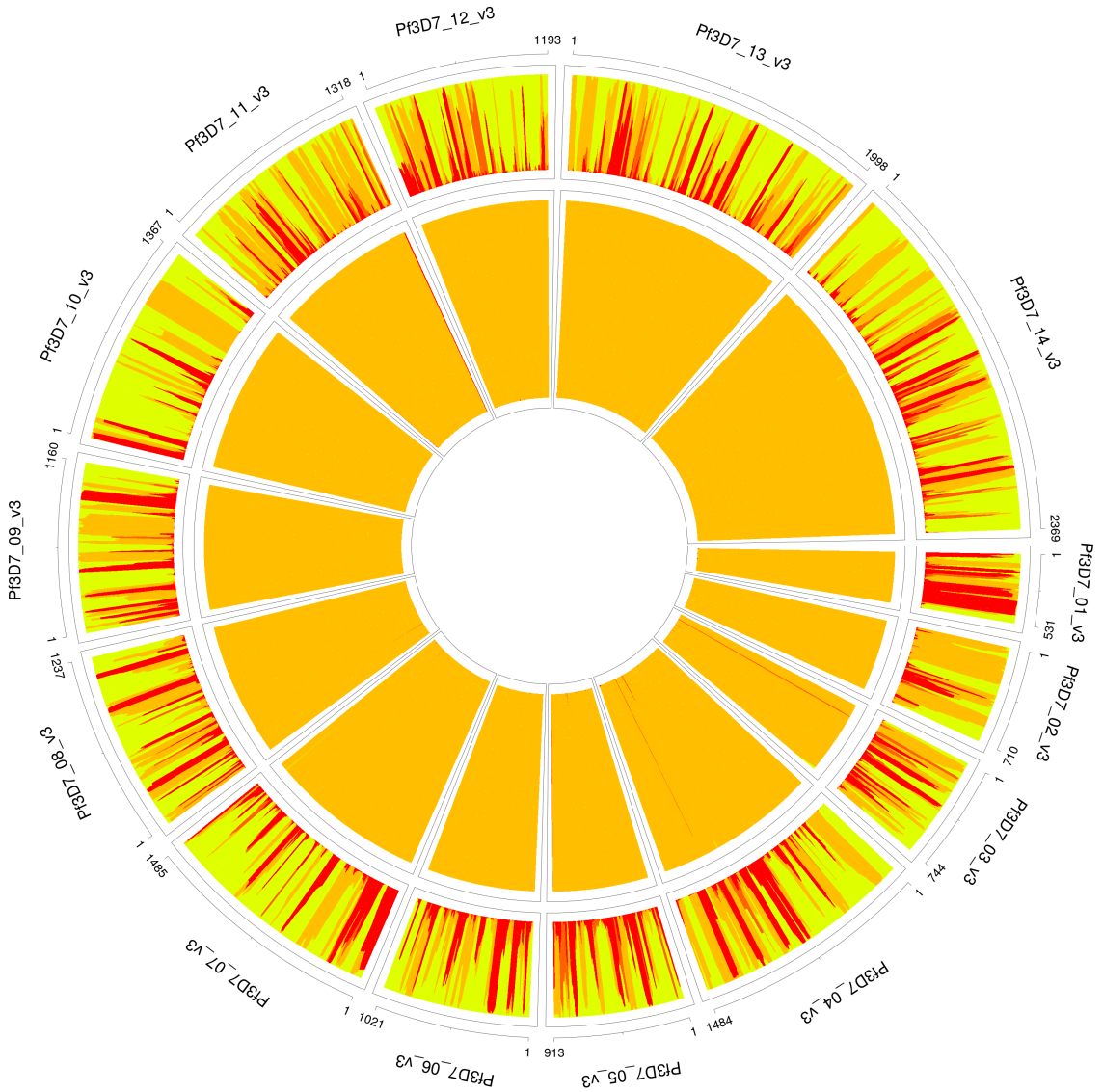
PG0400_sigma5

```
./dEploid -vcf data/exampleData/PG0400-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -panel  data/exampleData/labStrains.eg.panel.txt \
    -o PG0400-Csigma10 -seed 2 -sigma 10\
    -exclude exclude.txt

initialProp=$( cat PG0400-Csigma10.prop | tail -1 | sed -e "s/\t/ /g" )
./dEploid -vcf data/exampleData/PG0400-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -panel data/exampleData/labStrains.eg.panel.txt \
    -exclude exclude.txt \
    -o PG0400-Csigma10 \
    -initialP ${initialProp} \
    -painting PG0400-Csigma10.hap
```

(continues on next page)

(continued from previous page)

```
utilities/interpretDEploid.r -vcf data/exampleData/PG0400-C.eg.vcf.gz \
    -plaf data/exampleData/labStrains.eg.PLAF.txt \
    -dEprefix PG0400-Csigma10 \
    -o PG0400-Csigma10 \
    -exclude exclude.txt \
    -reverseRing -transformP
```



#PG0400_sigma1

7.3 Benchmark

Please refer to our paper Zhu et.al (2017) section 3 Validation and performance for benchmarking inference results on number of strains, proportions and haplotype quality.

For the enhanced version – DEploid-IBD, we compared our results against Zhu et.al (2017), and conducted more experiments and validations Zhu et.al (2019).

CHAPTER 8

Reporting Bugs

If you encounter any problem when using `dEploid`, please file a short bug report by using the issue tracker on GitHub or email joe.zhu (at) well.ox.ac.uk.

Please include the output of `dEploid -v` and the platform you are using `dEploid` on in the report. If the problem occurs while executing `dEploid`, please also include the command you are using and the random seed.

Thank you!

CHAPTER 9

Citing DEploid

If you use `dEploid` with the flag `-ibd`, please cite the following paper:

Zhu, J. S., J. A. Hendry, J. Almagro-Garcia, R. D. Pearson, R. Amato, A. Miles, D. J. Weiss, T. C. D. Lucas, M. Nguyen, P. W. Gething, D. Kwiatkowski, G. McVean, and for the Pf3k Project. (2018) The origins and relatedness structure of mixed infections vary with local prevalence of *P. falciparum* malaria. *biorxiv*, doi: https://doi.org/10.1101/387266.

Bibtex record::

```
@article {Zhu387266,
author = {Zhu, Sha Joe and Hendry, Jason A. and Almagro-Garcia, Jacob and Pearson,␣
↪Richard D. and Amato, Roberto and Miles, Alistair and Weiss, Daniel J. and Lucas,␣
↪Tim C.D. and Nguyen, Michele and Gething, Peter W. and Kwiatkowski, Dominic and␣
↪McVean, Gil and ,},
title = {The origins and relatedness structure of mixed infections vary with local␣
↪prevalence of P. falciparum malaria},
year = {2018},
doi = {10.1101/387266},
publisher = {Cold Spring Harbor Laboratory},
URL = {https://www.biorxiv.org/content/early/2018/08/09/387266},
eprint = {https://www.biorxiv.org/content/early/2018/08/09/387266.full.pdf},
journal = {bioRxiv}
}
```

If you use `dEploid` in your work, please cite the program:

Zhu, J. S. J. A. Garcia G. McVean. (2017) Deconvolution of multiple infections in *Plasmodium falciparum* from high throughput sequencing data. *Bioinformatics* btx530. doi: https://doi.org/10.1093/bioinformatics/btx530.

Bibtex record::

```
@article {Zhubtx530,
author = {Zhu, Sha Joe and Almagro-Garcia, Jacob and McVean, Gil},
title = {Deconvolution of multiple infections in {{\em Plasmodium falciparum}} from␣
↪high throughput sequencing data},
```

(continues on next page)

(continued from previous page)

```
year = {2017},
doi = {10.1093/bioinformatics/btx530},
URL = {https://doi.org/10.1093/bioinformatics/btx530},
journal = {Bioinformatics}
}
```

Bibliography

[Li2003] Li, N. and M. Stephens (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics* 165(4), 2213–2233.