Haplotype assembly in dipSPAdes

Svyatoslav Sidorov, St. Petersburg University of the RAS, Supervisor: Yana Safonova, Algorithmic Biology Lab, St. Petersburg University of the RAS
**dipSPAdes**

**dipSPAdes** is a new algorithm (and a tool) for assembling HP genomes that was developed at the Algorithmic Biology Lab (St. Petersburg University of the RAS).

It generates both consensus contigs and haplocontigs using de Bruijn graph.


http://link.springer.com/chapter/10.1007/978-3-319-05269-4_21

http://bioinf.spbau.ru
dipSPAdes pipeline

Adopted from: Yana Safonova, Anton Bankevich, Pavel A. Pevzner. *dipSPAdes: Assembler for Highly Polymorphic Diploid Genomes*
**dipSPAdes pipeline**

*Haplocontig* is a contig that was assembled from only one haplome.

Adopted from: **Yana Safonova, Anton Bankevich, Pavel A. Pevzner.** *dipSPAdes: Assembler for Highly Polymorphic Diploid Genomes.*
dipSPAdes pipeline

*Haplocontig* is a contig that was assembled from only one haplome.

1. Construct *diploid graph* from haplocontigs.

---

**dipSPAdes pipeline**

*Haplocontig* is a contig that was assembled from only one haplome.

1. Construct *diploid graph* from haplocontigs.
2. Transform *diploid graph* to *consensus graph* by an aggressive bulge collapsing.

Adopted from: **Yana Safonova, Anton Bankevich, Pavel A. Pevzner.** *dipSPAdes: Assembler for Highly Polymorphic Diploid Genomes.*
dipSPAdes pipeline

*Haplocontig* is a contig that was assembled from only one haplome.

1. Construct *diploid graph* from haplocontigs.
2. Transform *diploid graph* to *consensus graph* by an aggressive bulge collapsing.
3. *Mask polymorphisms in haplocontigs*, i. e. project haplocontigs on the consensus graph.

dipSPAdes pipeline

*Haplocontig* is a contig that was assembled from only one haplome.

1. Construct *diploid graph* from haplocontigs.
2. Transform *diploid graph* to *consensus graph* by an aggressive bulge collapsing.
3. *Mask polymorphisms in haplocontigs*, i.e. project haplocontigs on the consensus graph.
4. Construct *consensus contigs* by overlapping the masked haplocontigs (overlapping parts are the same).

Adopted from: **Yana Safonova, Anton Bankevich, Pavel A. Pevzner.** *dipSPAdes: Assembler for Highly Polymorphic Diploid Genomes.*
dipSPAdes pipeline

*Haplocontig* is a contig that was assembled from only one haplome.

1. Construct *diploid graph* from haplocontigs.
2. Transform *diploid graph* to *consensus graph* by an aggressive bulge collapsing.
3. Mask polymorphisms in *haplocontigs*, i.e. project haplocontigs on the consensus graph.
4. Construct *consensus contigs* by overlapping the masked haplocontigs (overlapping parts are the same).
5. Haplocontigs resolutions.

Problem (with haplotype assembly)

H1 ——> H2

edge of consensus graph

✓

(conservative region)

H1 ——> H2

edge of consensus graph

✓

(H1 and H2 are from different haplomes)

H1 ——> H2

edge of consensus graph

✓

(repeat?)
Project goals

1. Find out what mistakes dipSPAdes makes during the haplotype assembly.
2. Improve haplocontigs resolution and filtrate short haplocontigs with repeats.
3. Design a method for assembly of haplomes from resolved homologous haplocontigs.

(H1 and H3 are from the same haplome)
Pipeline for simulation of diploid datasets with known haplomes (based on a bunch of Python scripts):

- haplomes: first 100 Kbp from *E. coli* reference genome (GenBank: U00096.2) and its copy with 10 % of nucleotides randomly changed;
- from the two haplomes paired-end reads 100 bp long were ‘cutted’ with 50x average coverage and 250 bp insert size;
- reads obtained were assembled into haplocontigs with SPAdes (*-k 15 --diploid --only-assembler* options).

Reads from different haplomes can be stored in different FASTA files or in the same FASTA file.
Detection of SPAdes erroneous behavior: it may produce not only haplocontigs (as expected) but *chimeric contigs* too (contigs that contain polymorphic regions from both haplomes).

So we proceed to simulation of an ‘ideal’ haplocontigs assembly (we assembled reads from each haplome separately).
Result 3

An idea of haplocontigs resolution improvement and filtration of haplocontigs with repeats:

- construct a bipartite *conflict graph* where vertices are haplocontigs, and they’re connected iff dipSPAdes says that they belong to different haplomes;
- so, for every connected component of the conflict graph we know what haplome each haplocontig belongs to.
How to construct *bipartite* conflict graph?

**Algorithm sketch:**
- dipSPAdes can provide us with individual edges of the conflict graph;
- we should assign some weight to each edge so that it’s big if two haplocontigs belong to different haplomes and small if they belong to one and the same haplome;
- keep adding edges with max weight to the graph until it ceases to be bipartite.
Conflict graph

How to assign weights to edges?

The weight is (ideas we tested):

- a number of consensus graph edges on which two haplocontigs have a subsequence in common;
- a total length of consensus graph edges on which two haplocontigs have a subsequence in common;
- a number of bulges and shared edges in diploid graph that two subsequences have in common.
**Edge weight**

**Idea 1**: weight is a number of consensus graph edges on which two haplocontigs have a subsequence in common.
**Edge weight**

**Idea 1:** weight is a number of consensus graph edges on which two haplocontigs have a subsequence in common.

When the graph was constructed we lost the right edge \{3, 6\} but got the wrong edge \{9, 10\}. 

**Edge weight**

**Idea 2:** weight is a total length of consensus graph edges on which two haplocontigs have a subsequence in common.
Edge weight

**Idea 2:** weight is a total length of consensus graph edges on which two haplocontigs have a subsequence in common.
**Edge weight**

**Idea 3:** weight is a number of bulges and shared edges in diploid graph that two subsequences have in common.
Edge weight

Idea 3: weight is a number of bulges and shared edges in diploid graph that two subsequences have in common.

There are only right edges in the conflict graph.

weight = (# of bulges, # of shared edges)
Results

- pipeline for simulation of diploid datasets with known haplomes was developed (a bunch of Python scripts were written);
- SPAdes erroneous behavior was detected: it may produce not only haplocontigs (as expected) but chimeric contigs;
- algorithm for haplotype assembly and haplocontigs filtration was developed and partially implemented in dipSPAdes.
Thank you!

*Clavelina moluccensis*, a tunicate. Tunicates are diploids with high polymorphism rate.