# Applied Genome Research

# Assembly & Scaffolding

205048 & 205049

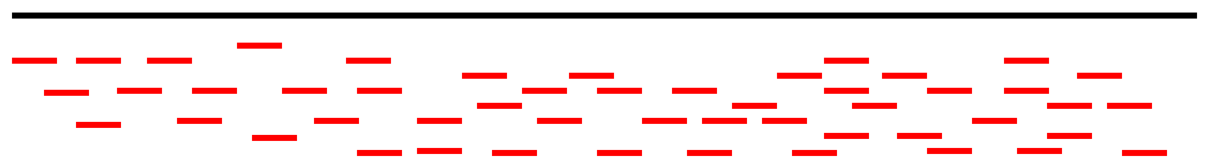
Boas Pucker

# Overview

- Assembly theory
- SOAPdenovo2
- Assembly evaluation
- Scaffolding theory
- SSPACE

# Assembly problem

- Genome sequence length exceeds read length!



Genome sequence (125Mbp)

Illumina Reads (32-300nt)

Coverage (read coverage depth) = number of reads at a certain base in the genome

# EXERCISE

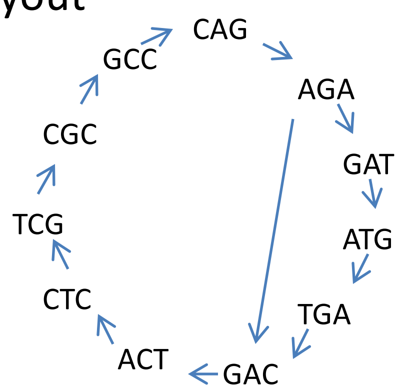
- Calculate the average coverage of SRX…… for the Col-0 reference genome sequence (120Mbp)!

# Assembly theory

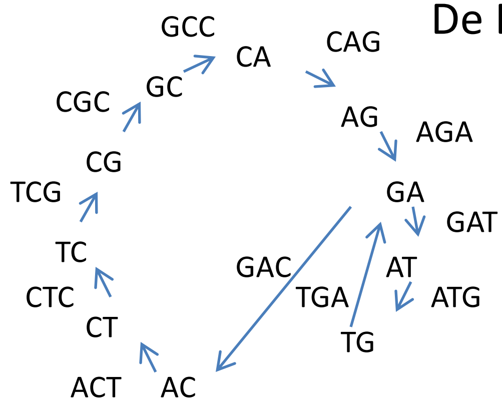
Genome: CAGATGACTCG

```
CAG
 AGA
  GAT
   ATG
    …
```

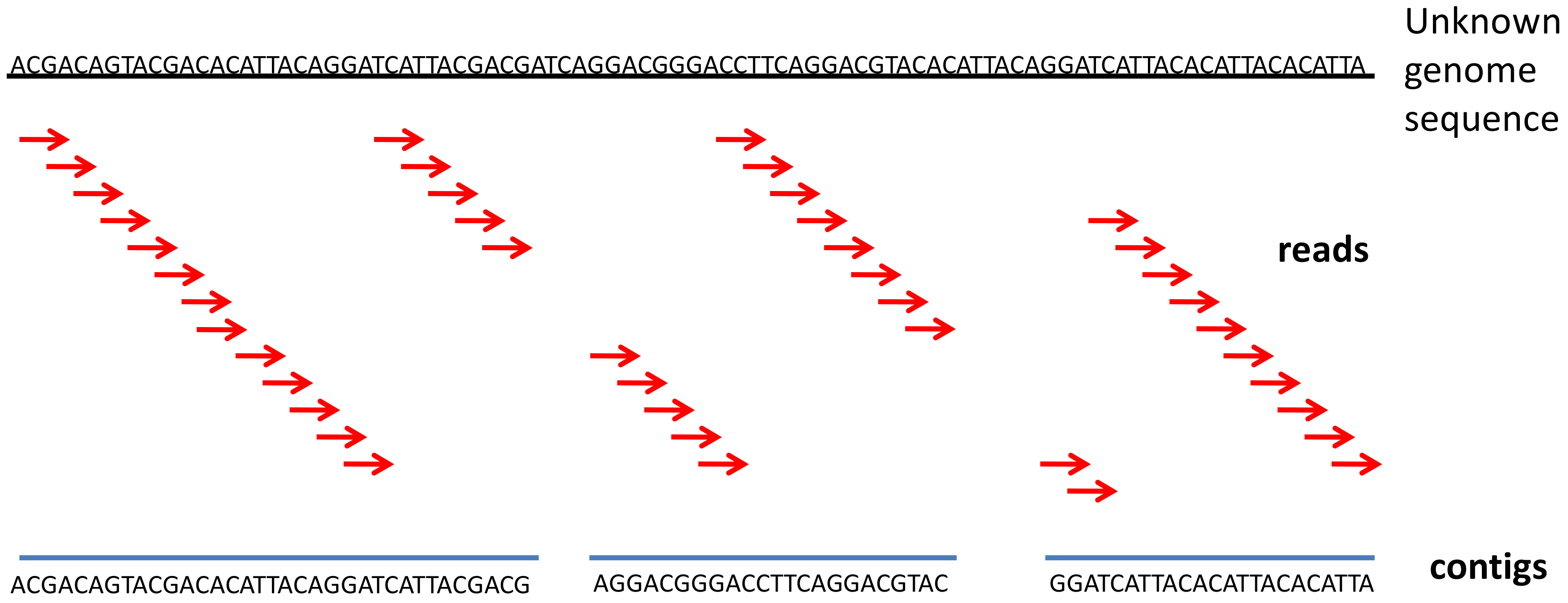
Overlap layout consensus (OLC)



De Bruijn graph

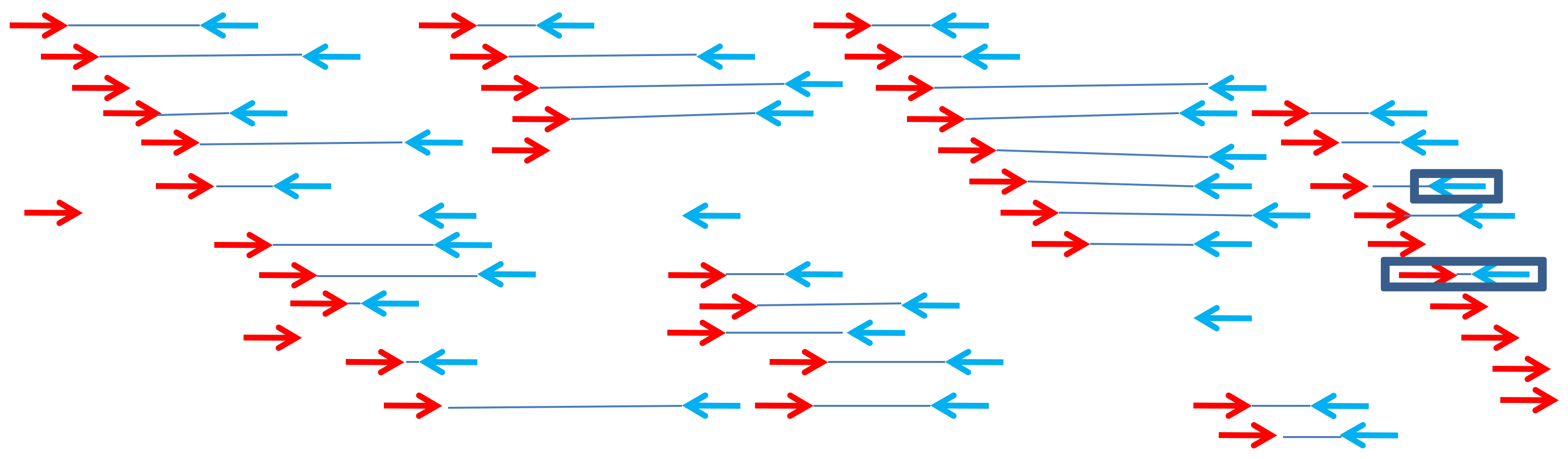


# Assembly


ACGACAGTACGACACATTACAGGATCATTACGACGATCAGGACGGGACCTTCAGGACGTACACATTACAGGATCATTACACATTACACATTA
Unknown genome sequence
reads
contigs
ACGACAGTACGACACATTACAGGATCATTACGACG
AGGACGGGACCTTCAGGACGTAC
GGATCATTACACATTACACATTA

# Assembly

ACGACAGTACGACACATTACAGGATCATTACGACGATCAGGACGGGACCTTCAGGACGTACACATTACAGGATCATTACACATTACACATTA Unknown genome sequence



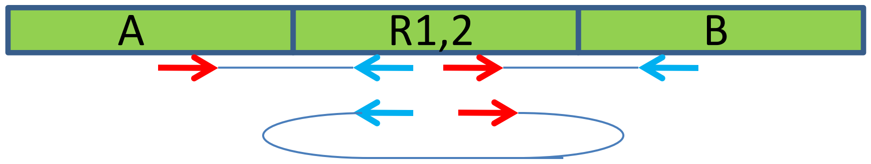
ACGACAGTACGACACATTACAGGATCATTACGACG AGGACGGGACCTTCAGGACGTAC GGATCATTACACATTACACATTA **contig**

ACGACAGTACGACACATTACAGGATCATTACGACGNNNNAGGACGGGACCTTCAGGACGTACNNNNNNGGATCATTACACATTACACATTA **scaffold**

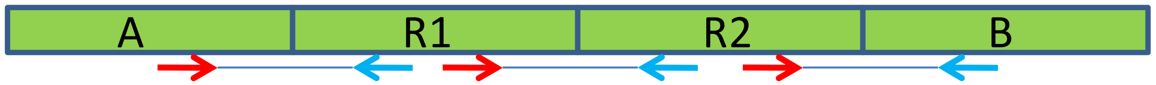
Contigs are connected by spanning fragments into scaffold: approximate size of gaps is known, but sequence remains unknown!

# Assembly issues

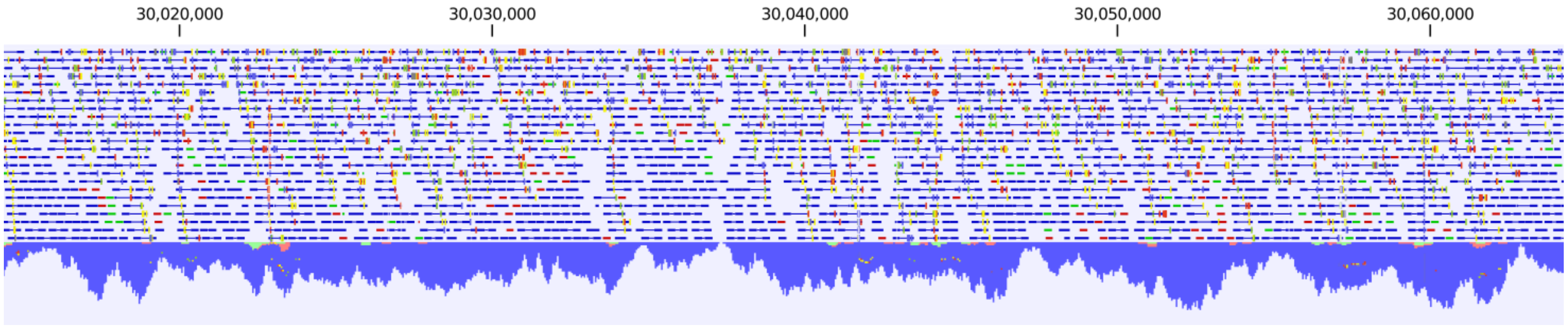
Miss-assembly:



Correct assembly:



# Assembly issues II

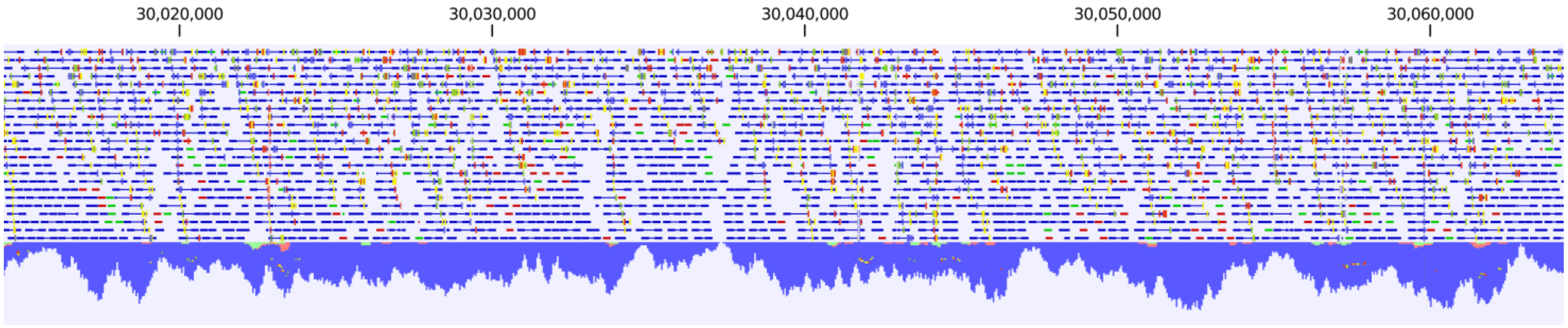


Read mapping of paired-end sequenced fragments (blue) to assembly

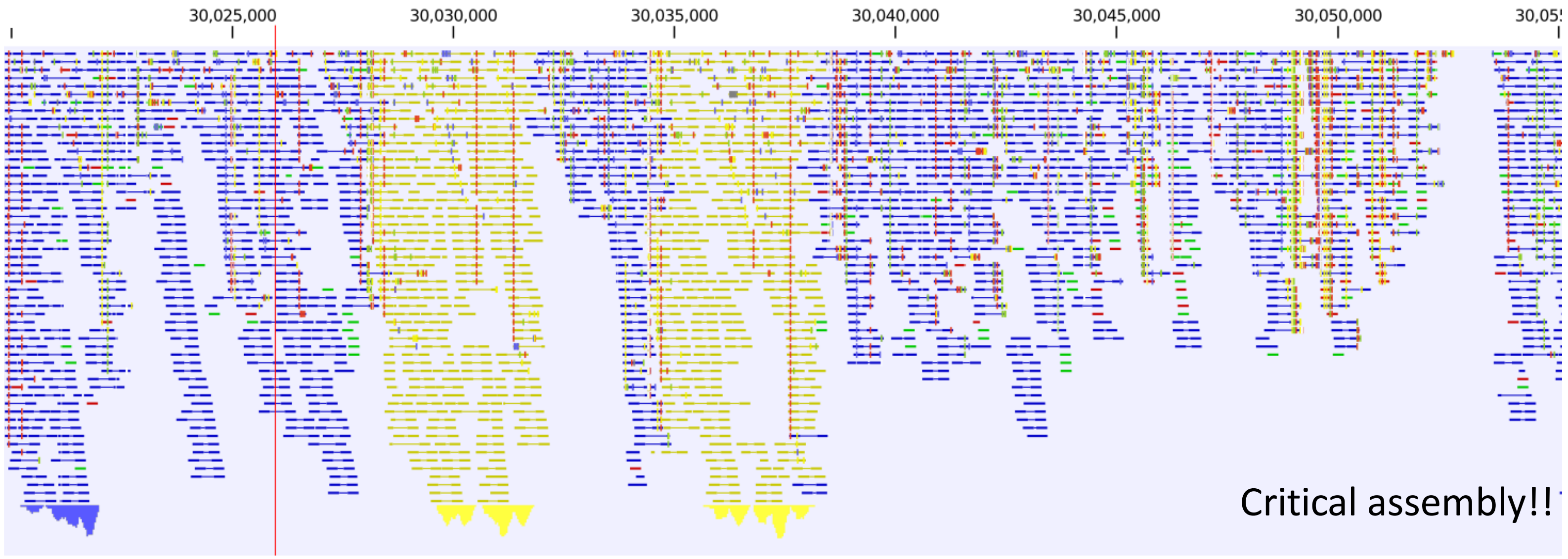
Coverage is too high to show all individual fragments at some positions

Does it look like a good assembly?

# Assembly issues II



Read mapping of paired-end sequenced fragments (blue) to assembly



Critical assembly!!

## SOAPdenovo2

- Different “versions” available (<63bp kmers, <127bp kmers)
- Input data cannot be compressed
- One of the best NGS assemblers for heterozygous organisms
- Includes scaffolding (we will use SSPACE for this)

# SOAPdenovo2 – config file

- max_rd_len = maximal read length
- avg_ins = average insert size (distance between paired reads)
- asm_flags = number of part in process to use this reads
- rd_len_cutoff = specifies length of reads to use
- rank = importance of this data set
- pair_num_cutoff = number of read pairs to connect contigs for scaffolding
- map_len = minimal length of alignment during read mapping
- q1 = fastq file with forward reads
- q2 = fastq file with reverse reads

# SOAPdenovo2 - usage

- Example:

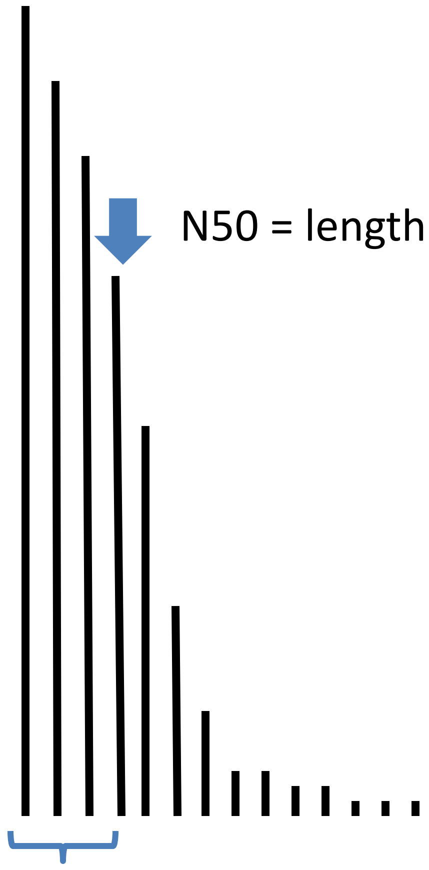
```
$ SOAPdenovo-63mer all –s example_config.txt -K 63 -R -p 4 -o
  ./first_test 2>assembly.log 1>assembly.err
```

- All … runs all parts of assembly process
- -s <CONFIG_FILE> … information about data are provided in file
- -K <INT> … k-mer size (<=63)
- -R … try to resolve repeats
- -p … number of CPUs to use for assembly
- -o … prefix for results to save
- 1> …. error log file
- 2> …. output log file

# EXERCISE

- Run SOAPdenovo2 assembly!

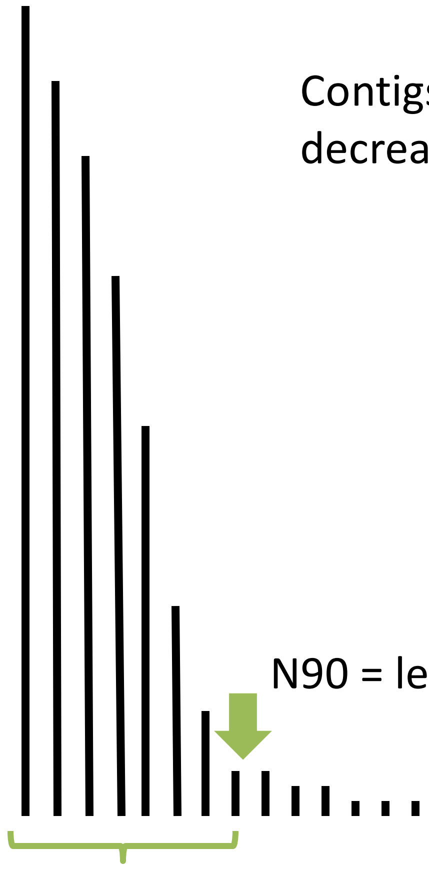
# Assembly evaluation – Nx for continuity quantification



Contigs are sorted by decreasing size.

Contigs sum up to 50% of total assembly size

# Assembly evaluation – Nx for continuity quantification



Contigs are sorted by decreasing size.

N90 = length of this contig

Contigs sum up to 90% of total assembly size

## EXERCISE

- Run python script to analyze results:

```
$ python contig_stats.py - -input <filename>
```

# Assembly evaluation – read mapping for completeness quantification

- Percentage of mapped reads indicates quality of assembly
- Due to sequencing errors and different artifacts the mapping rate will always be lower than 100%
- Very low and very high coverage regions might be missing
- Mapping of paired-end or mate pair reads can indicate rearrangements in the assembly (e.g. REAPR)
- BUSCO analysis would be another possibility:
  - Generally conserved genes in the assembly are counted
  - (not applicable in our example, because the assembled contigs represent only a small fraction of the genome sequence)

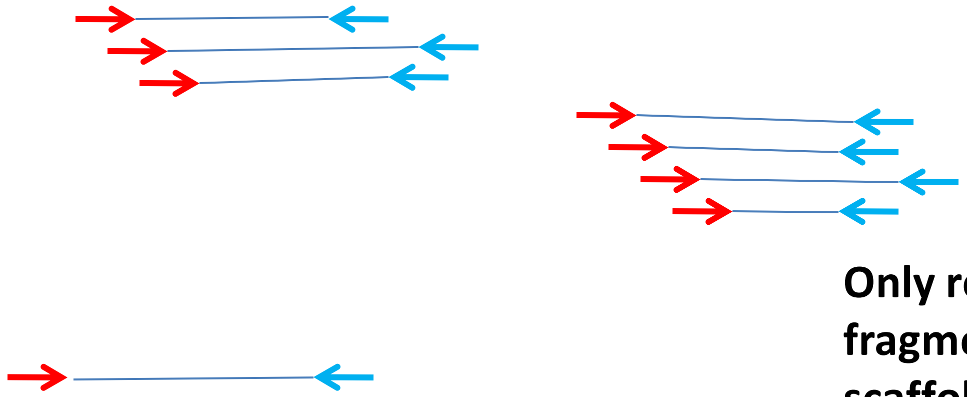
## QUESTION

- How is it possible to increase the continuity of an assembly?

# Scaffolding - theory

ACGACAGTACGACACATTACAGGATCATTACGACGATCAGGACGGGACCTTCAGGACGTACACATTACAGGATCATTACACATTACACATTA

Unknown genome sequence



**Only reads of contig connecting fragments are used for scaffolding.**

ACGACAGTACGACACATTACAGGATCATTACGACG AGGACGGGACCTTCAGGACGTAC GGATCATTACACATTACACATTA

ACGACAGTACGACACATTACAGGATCATTACGACGNNNNAGGACGGGACCTTCAGGACGTACNNNNNNGGATCATTACACATTACACATTA

Contigs are connected by spanning fragments into scaffold: approximate size of gaps is known, but sequence remains unknown!

# Scaffolding - SSPACE

$ SSPACE_Standard_v3.0.pl\

-l <TEXTFILE>\ …. File contains information about reads

-s <ASSEMBLY>\ … SOAP assembly result file (contigs)

-k <NUMBER_OF_LINKS>\ … number of linking fragments to connect contigs

-T <NUMBER_OF_THREADS>\ … number of threads to use

-b <BASE_NAME> … output prefix (only small string required)

## EXERCISE

- Run SSPACE with the SOAPdenovo2 assembled contigs!

## Scaffolding - evaluation

- Continuity = length distribution of scaffolds compared to length distribution of contigs
- Correctness = rearrangements can be identified by mapping paired reads (errors caused by overscaffolding)

# EXERCISE

- Calculate the sequence length distribution stats of the scaffolds (contig_stats.py)!
- Compare results of SOAP contigs, SOAP scaffolds and SSPACE scaffolds!
- Search the literature for values (e.g. N50 or genome size) and compare them to your own results!