HG-CoLoR: Hybrid Graph for the error Correction of Long Reads

Pierre Morisse, Thierry Lecroq and Arnaud Lefebvre
pierre.morisse2@univ-rouen.fr

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes

July 5, 2017

Plan

Next Generation Sequencing

- In 2005, Next Generation Sequencing (NGS) technologies started to develop
- Production of millions of short sequences (100-300 bases), called reads, used to resolve mapping and assembly problems
- Due to their high number, efficient algorithms are required to process these reads
- These reads also contain sequencing errors ($\sim 1\%$)
- NGS data analysis became an important research field

Next Generation Sequencing

- In 2005, Next Generation Sequencing (NGS) technologies started to develop
- Production of millions of short sequences (100-300 bases), called reads, used to resolve mapping and assembly problems
- Due to their high number, efficient algorithms are required to process these reads
- These reads also contain sequencing errors ($\sim 1\%$)
- NGS data analysis became an important research field

Next Generation Sequencing

- In 2005, Next Generation Sequencing (NGS) technologies started to develop
- Production of millions of short sequences (100-300 bases), called reads, used to resolve mapping and assembly problems
- Due to their high number, efficient algorithms are required to process these reads
- These reads also contain sequencing errors ($\sim 1\%$)
- NGS data analysis became an important research field

Next Generation Sequencing

- In 2005, Next Generation Sequencing (NGS) technologies started to develop
- Production of millions of short sequences (100-300 bases), called reads, used to resolve mapping and assembly problems
- Due to their high number, efficient algorithms are required to process these reads
- These reads also contain sequencing errors ($\sim$ 1%)
- NGS data analysis became an important research field

Next Generation Sequencing

- In 2005, Next Generation Sequencing (NGS) technologies started to develop
- Production of millions of short sequences (100-300 bases), called reads, used to resolve mapping and assembly problems
- Due to their high number, efficient algorithms are required to process these reads
- These reads also contain sequencing errors ($\sim$ 1%)
- NGS data analysis became an important research field

Third Generation Sequencing

- More recently, Third Generation Sequencing technologies started to develop
- Two main technologies: Pacific Biosciences and Oxford Nanopore
- Allow the sequencing of longer reads (several thousand of bases)
- Very useful to resolve assembly problems for large and complex genomes
- Much higher error rate, around 15% for Pacific Biosciences and up to 30% for Oxford Nanopore

Third Generation Sequencing

- More recently, Third Generation Sequencing technologies started to develop
- Two main technologies: Pacific Biosciences and Oxford Nanopore
- Allow the sequencing of longer reads (several thousand of bases)
- Very useful to resolve assembly problems for large and complex genomes
- Much higher error rate, around 15% for Pacific Biosciences and up to 30% for Oxford Nanopore

Third Generation Sequencing

- More recently, Third Generation Sequencing technologies started to develop
- Two main technologies: Pacific Biosciences and Oxford Nanopore
- Allow the sequencing of longer reads (several thousand of bases)
- Very useful to resolve assembly problems for large and complex genomes
- Much higher error rate, around 15% for Pacific Biosciences and up to 30% for Oxford Nanopore

Third Generation Sequencing

- More recently, Third Generation Sequencing technologies started to develop
- Two main technologies: Pacific Biosciences and Oxford Nanopore
- Allow the sequencing of longer reads (several thousand of bases)
- Very useful to resolve assembly problems for large and complex genomes
- Much higher error rate, around 15% for Pacific Biosciences and up to 30% for Oxford Nanopore

Third Generation Sequencing

- More recently, Third Generation Sequencing technologies started to develop
- Two main technologies: Pacific Biosciences and Oxford Nanopore
- Allow the sequencing of longer reads (several thousand of bases)
- Very useful to resolve assembly problems for large and complex genomes
- Much higher error rate, around 15% for Pacific Biosciences and up to 30% for Oxford Nanopore

Problem

- Due to their high error rate, error correction of long reads is mandatory
- Various methods already exist for the correction of short reads, but are not applicable to long reads
- Forces the development of new error correction methods
- Two main categories: self-correction and hybrid correction

Problem

- Due to their high error rate, error correction of long reads is mandatory
- Various methods already exist for the correction of short reads, but are not applicable to long reads
- Forces the development of new error correction methods
- Two main categories: self-correction and hybrid correction

Problem

- Due to their high error rate, error correction of long reads is mandatory
- Various methods already exist for the correction of short reads, but are not applicable to long reads
- Forces the development of new error correction methods
- Two main categories: self-correction and hybrid correction

Problem

- Due to their high error rate, error correction of long reads is mandatory
- Various methods already exist for the correction of short reads, but are not applicable to long reads
- Forces the development of new error correction methods
- Two main categories: self-correction and hybrid correction

Inspiration

- NaS [Madoui et al., 2015]
- Does not locally correct erroneous regions
- Uses long reads as templates to generate corrected long reads from assemblies of short reads
- Requires the mapping of the short reads both on the long reads and against each other

Inspiration

- NaS [Madoui et al., 2015]
- Does not locally correct erroneous regions
- Uses long reads as templates to generate corrected long reads from assemblies of short reads
- Requires the mapping of the short reads both on the long reads and against each other

Inspiration

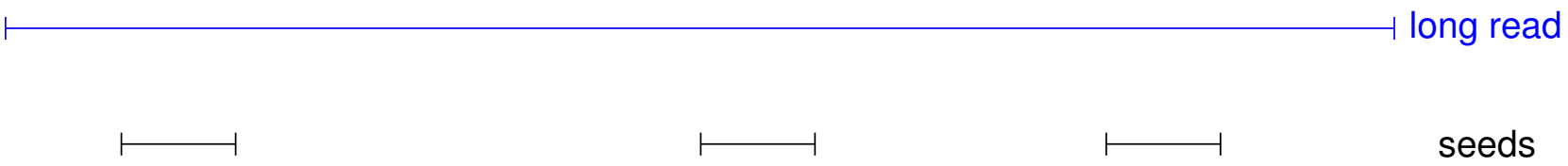
- NaS [Madoui et al., 2015]
- Does not locally correct erroneous regions
- Uses long reads as templates to generate corrected long reads from assemblies of short reads
- Requires the mapping of the short reads both on the long reads and against each other

Inspiration

- NaS [Madoui et al., 2015]
- Does not locally correct erroneous regions
- Uses long reads as templates to generate corrected long reads from assemblies of short reads
- Requires the mapping of the short reads both on the long reads and against each other

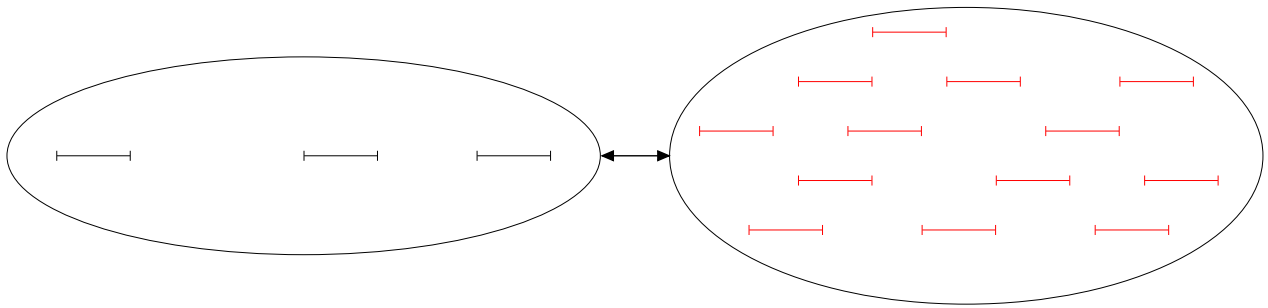
NaS overview

NaS corrects a long read as follows:



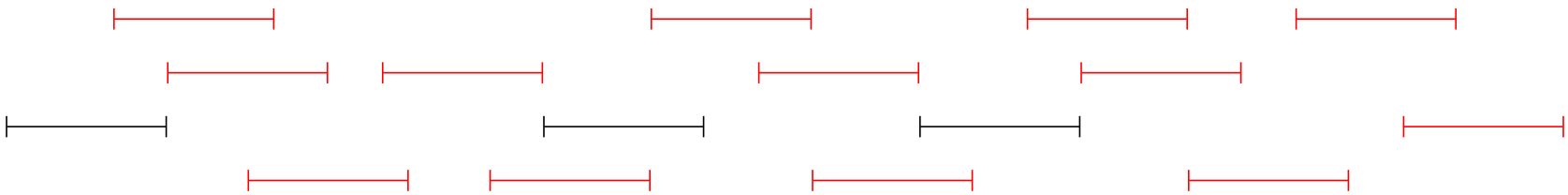
NaS overview

NaS corrects a long read as follows:



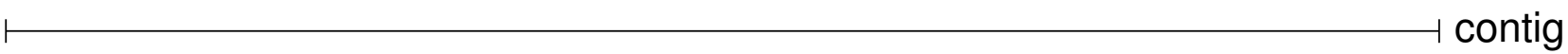
NaS overview

NaS corrects a long read as follows:



NaS overview

NaS corrects a long read as follows:



Main idea

- Generate corrected long reads from assemblies of short reads
- Get rid of the time consuming step of aligning the short reads against each other
- Focus on a seed and extend approach
- Rely on a hybrid structure between a de Bruijn graph and an overlap graph, built from the short reads

Main idea

- Generate corrected long reads from assemblies of short reads
- Get rid of the time consuming step of aligning the short reads against each other
- Focus on a seed and extend approach
- Rely on a hybrid structure between a de Bruijn graph and an overlap graph, built from the short reads

Main idea

- Generate corrected long reads from assemblies of short reads
- Get rid of the time consuming step of aligning the short reads against each other
- Focus on a seed and extend approach
- Rely on a hybrid structure between a de Bruijn graph and an overlap graph, built from the short reads

Hybrid graph

Overlap graph

de Bruijn graph

Idea

Mix the advantages of a de Bruijn graph and of an overlap graph, and allow to compute overlaps of variable lengths between the k-mers from the reads of a given set.

Hybrid graph

Overlap graph

de Bruijn graph

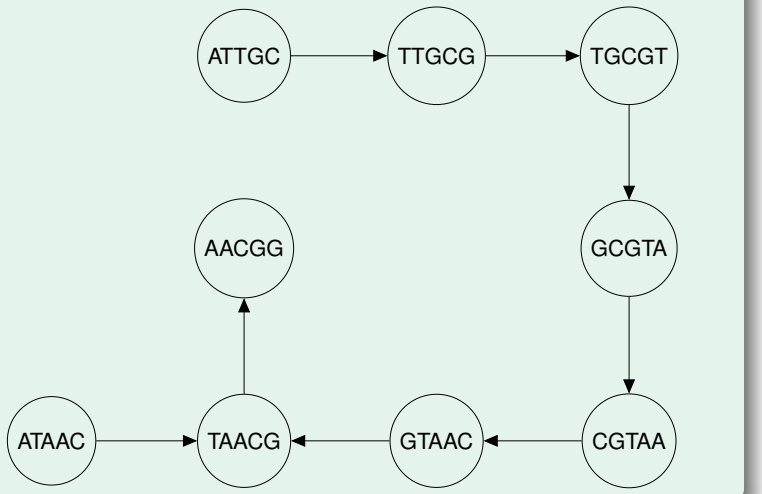
Idea

Mix the advantages of a de Bruijn graph and of an overlap graph, and allow to compute overlaps of variable lengths between the k-mers from the reads of a given set.

Hybrid graph

Overlap graph

de Bruijn graph



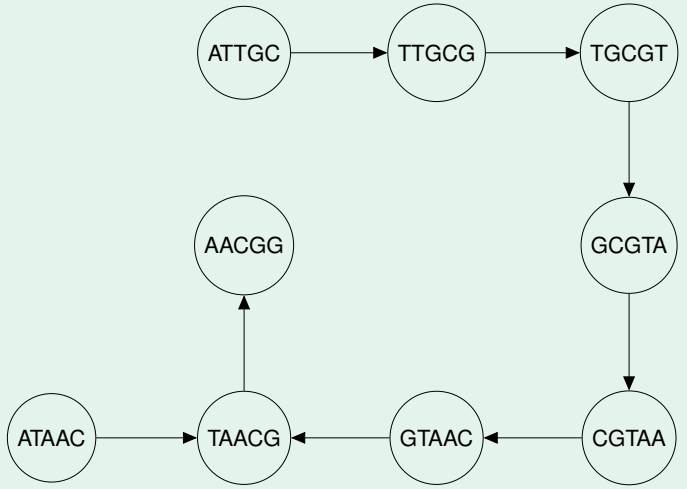
Idea

Mix the advantages of a de Bruijn graph and of an overlap graph, and allow to compute overlaps of variable lengths between the k-mers from the reads of a given set.

Hybrid graph

Overlap graph

de Bruijn graph



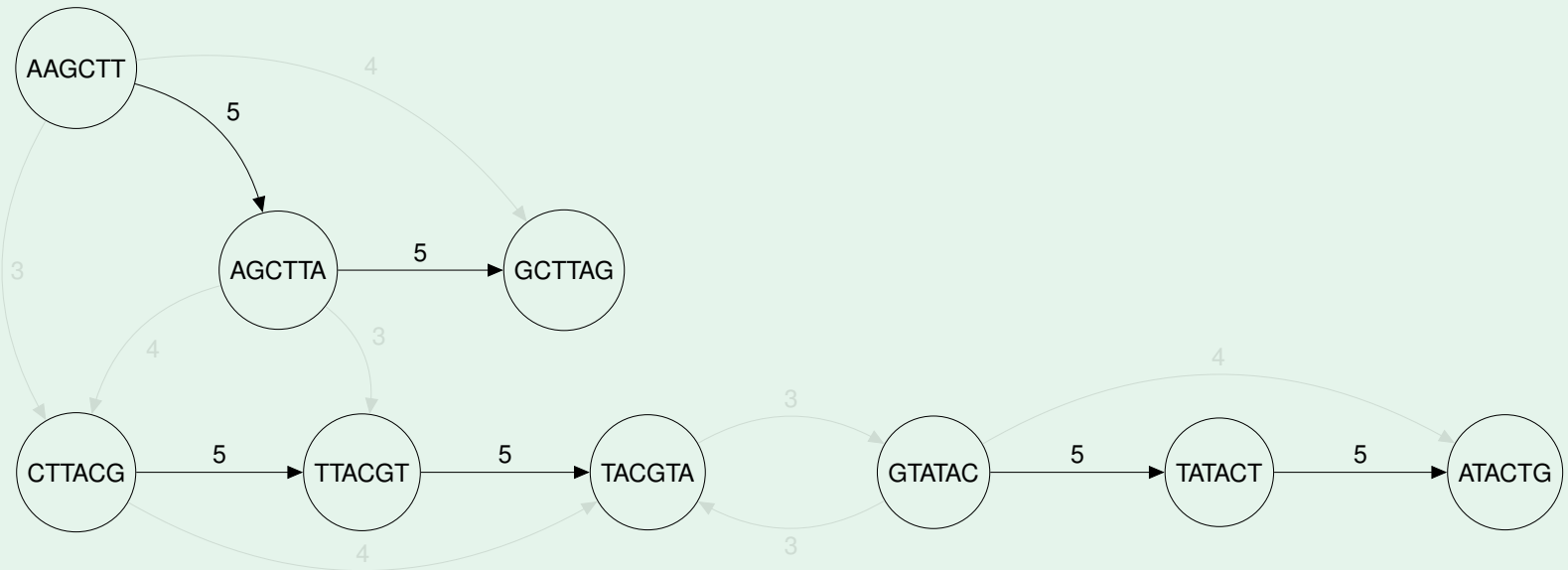
Idea

Mix the advantages of a de Bruijn graph and of an overlap graph, and allow to compute overlaps of variable lengths between the *k*-mers from the reads of a given set.

Hybrid graph

Example

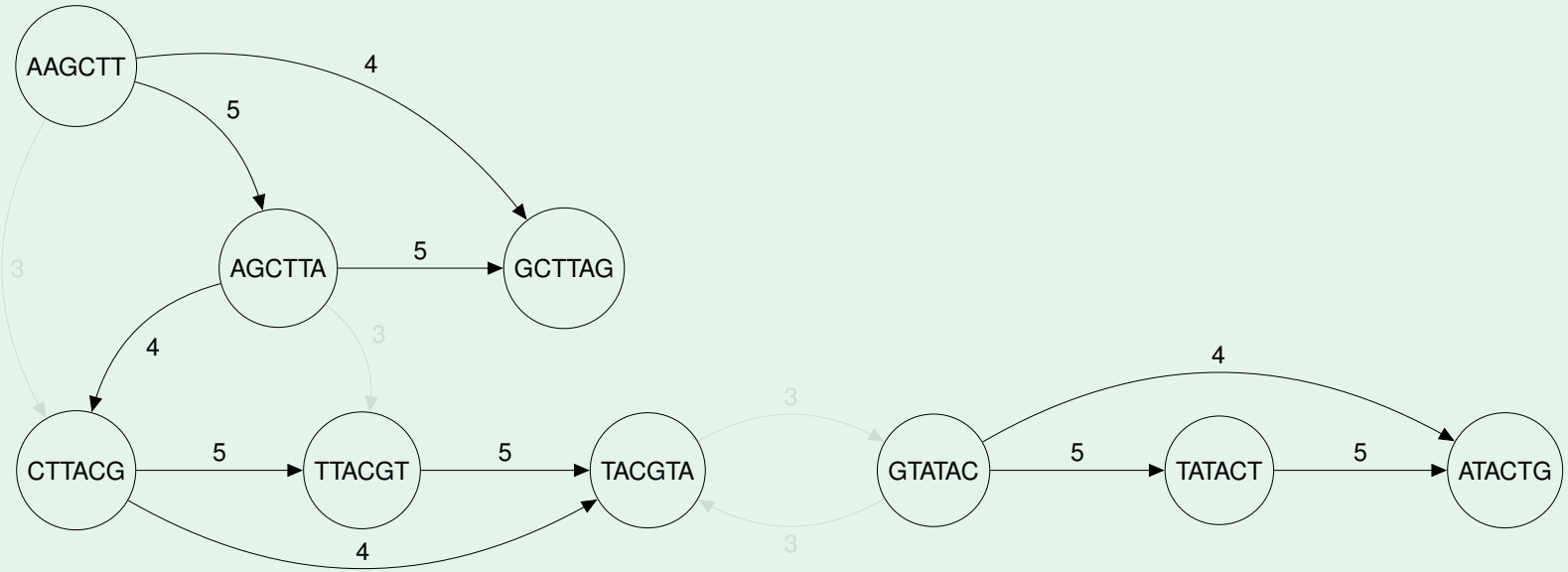
On the set of reads S = {AAGCTTAG, CTTACGTA, GTATACTG}



Hybrid graph

Example

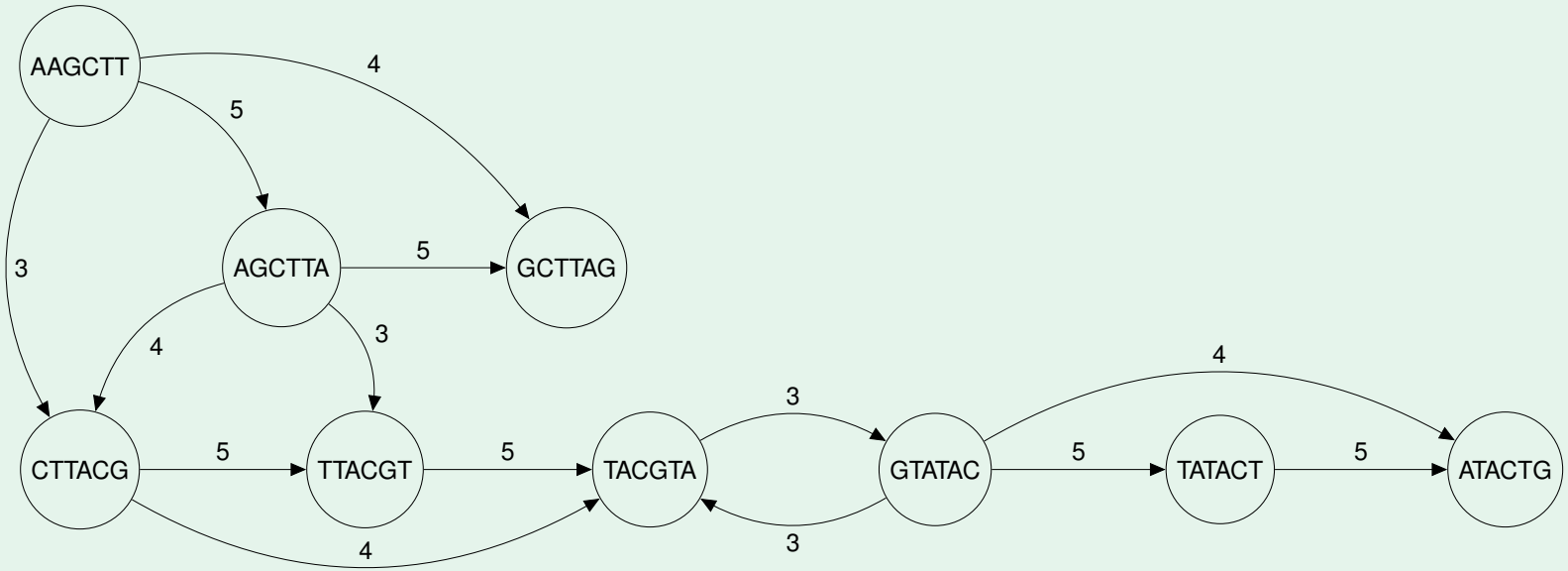
On the set of reads S = {AAGCTTAG, CTTACGTA, GTATACTG}



Hybrid graph

Example

On the set of reads S = {AAGCTTAG, CTTACGTA, GTATACTG}



Hybrid graph traversal

- The graph is not explicitly built
- Its traversal is simulated with PgSA [Kowalski et al., 2015]
- PgSA can index a set of reads and answer queries about strings of variable lengths
- One of the queries returns the positions of all the occurrences of a given string in the different reads

Hybrid graph traversal

- The graph is not explicitly built
- Its traversal is simulated with PgSA [Kowalski et al., 2015]
- PgSA can index a set of reads and answer queries about strings of variable lengths
- One of the queries returns the positions of all the occurrences of a given string in the different reads

Hybrid graph traversal

- The graph is not explicitly built
- Its traversal is simulated with PgSA [Kowalski et al., 2015]
- PgSA can index a set of reads and answer queries about strings of variable lengths
- One of the queries returns the positions of all the occurrences of a given string in the different reads

Hybrid graph traversal

- The graph is not explicitly built
- Its traversal is simulated with PgSA [Kowalski et al., 2015]
- PgSA can index a set of reads and answer queries about strings of variable lengths
- One of the queries returns the positions of all the occurrences of a given string in the different reads

Workflow

5 steps:

1. Correct the short reads
2. Align the short reads on the long read, to find seeds
3. Merge the overlapping seeds
4. Link the seeds, by traversing the hybrid graph
5. Extend the obtained corrected long read, on the left (resp. right) of the leftmost (resp. rightmost) seed

Workflow

5 steps:

1. Correct the short reads
2. Align the short reads on the long read, to find seeds
3. Merge the overlapping seeds
4. Link the seeds, by traversing the hybrid graph
5. Extend the obtained corrected long read, on the left (resp. right) of the leftmost (resp. rightmost) seed

Workflow

5 steps:

1. Correct the short reads
2. Align the short reads on the long read, to find seeds
3. Merge the overlapping seeds
4. Link the seeds, by traversing the hybrid graph
5. Extend the obtained corrected long read, on the left (resp. right) of the leftmost (resp. rightmost) seed

Workflow

5 steps:

1. Correct the short reads
2. Align the short reads on the long read, to find seeds
3. Merge the overlapping seeds
4. Link the seeds, by traversing the hybrid graph
5. Extend the obtained corrected long read, on the left (resp. right) of the leftmost (resp. rightmost) seed

Workflow

5 steps:

1. Correct the short reads
2. Align the short reads on the long read, to find seeds
3. Merge the overlapping seeds
4. Link the seeds, by traversing the hybrid graph
5. Extend the obtained corrected long read, on the left (resp. right) of the leftmost (resp. rightmost) seed

Step 4: Seeds linking

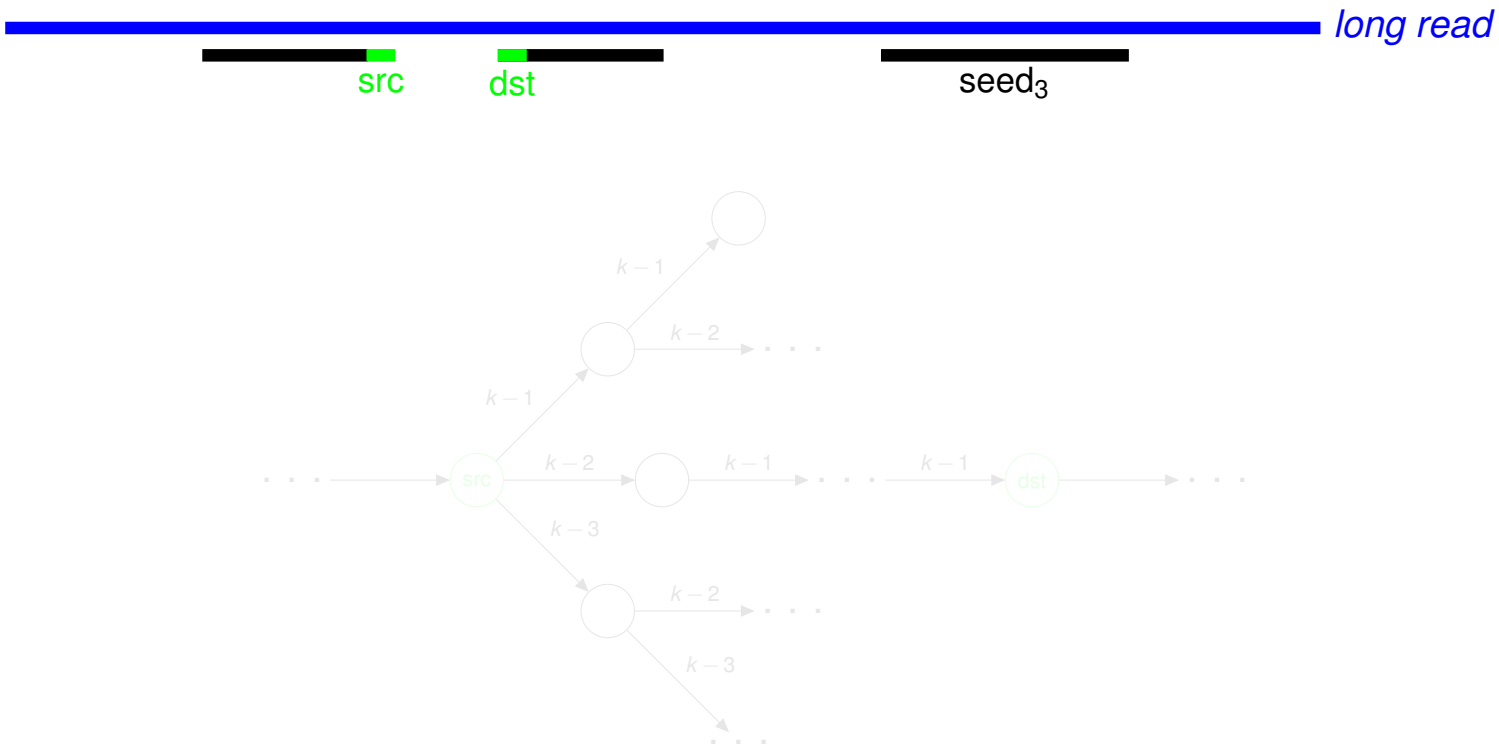
- Seeds are used as anchor points on the hybrid graph
- The graph is traversed to link together the seeds and assemble the *k*-mers

Step 4: Seeds linking

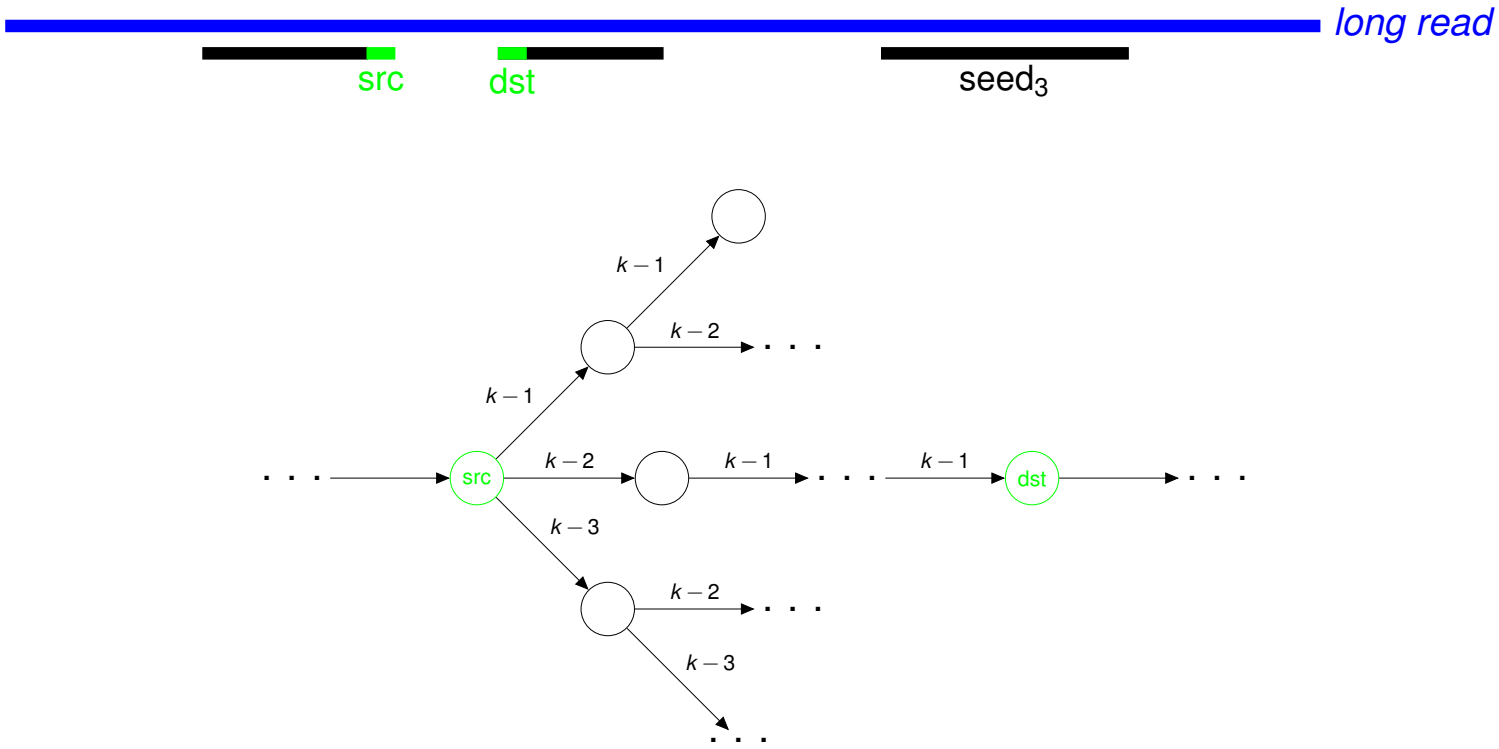
- Seeds are used as anchor points on the hybrid graph
- The graph is traversed to link together the seeds and assemble the *k*-mers

Step 4: Seeds linking

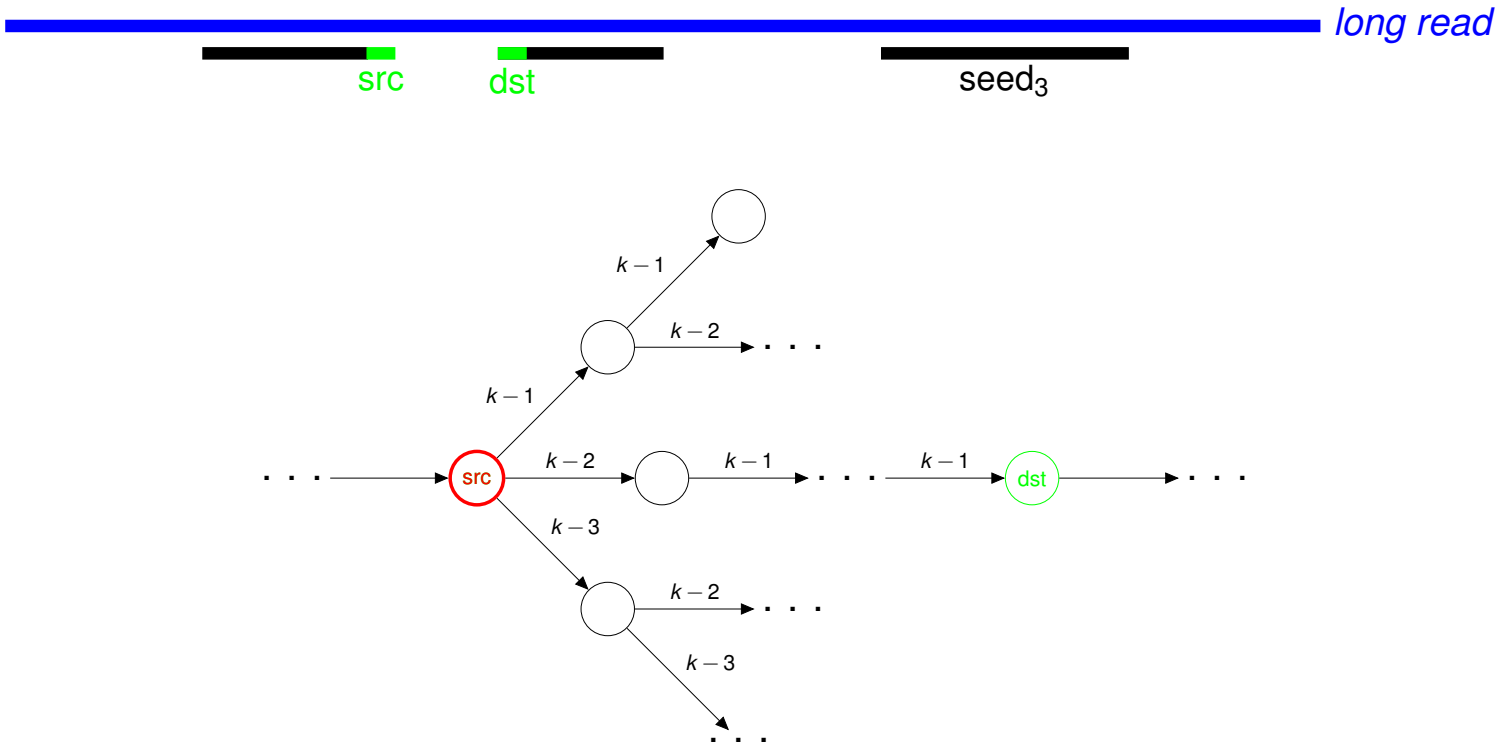
Step 4: Seeds linking



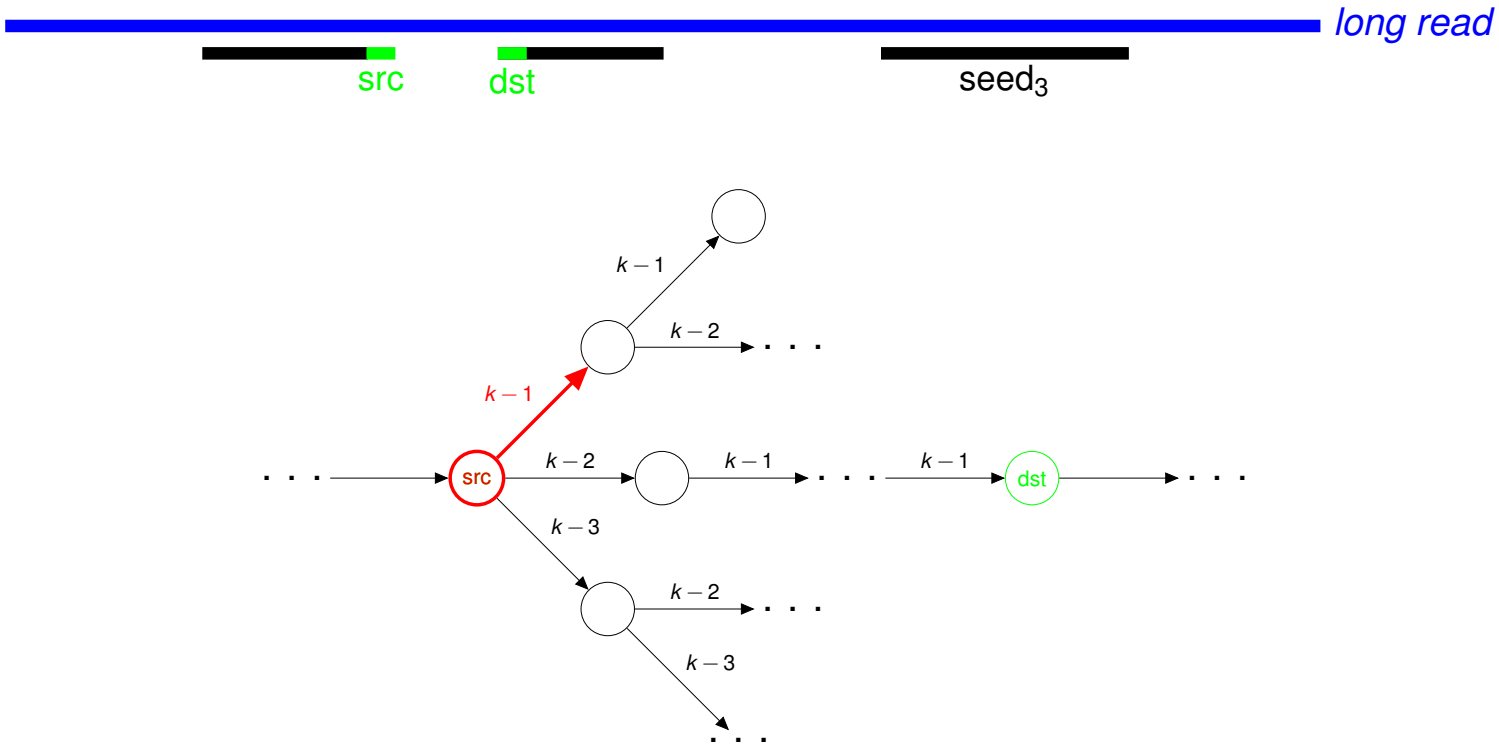
Step 4: Seeds linking



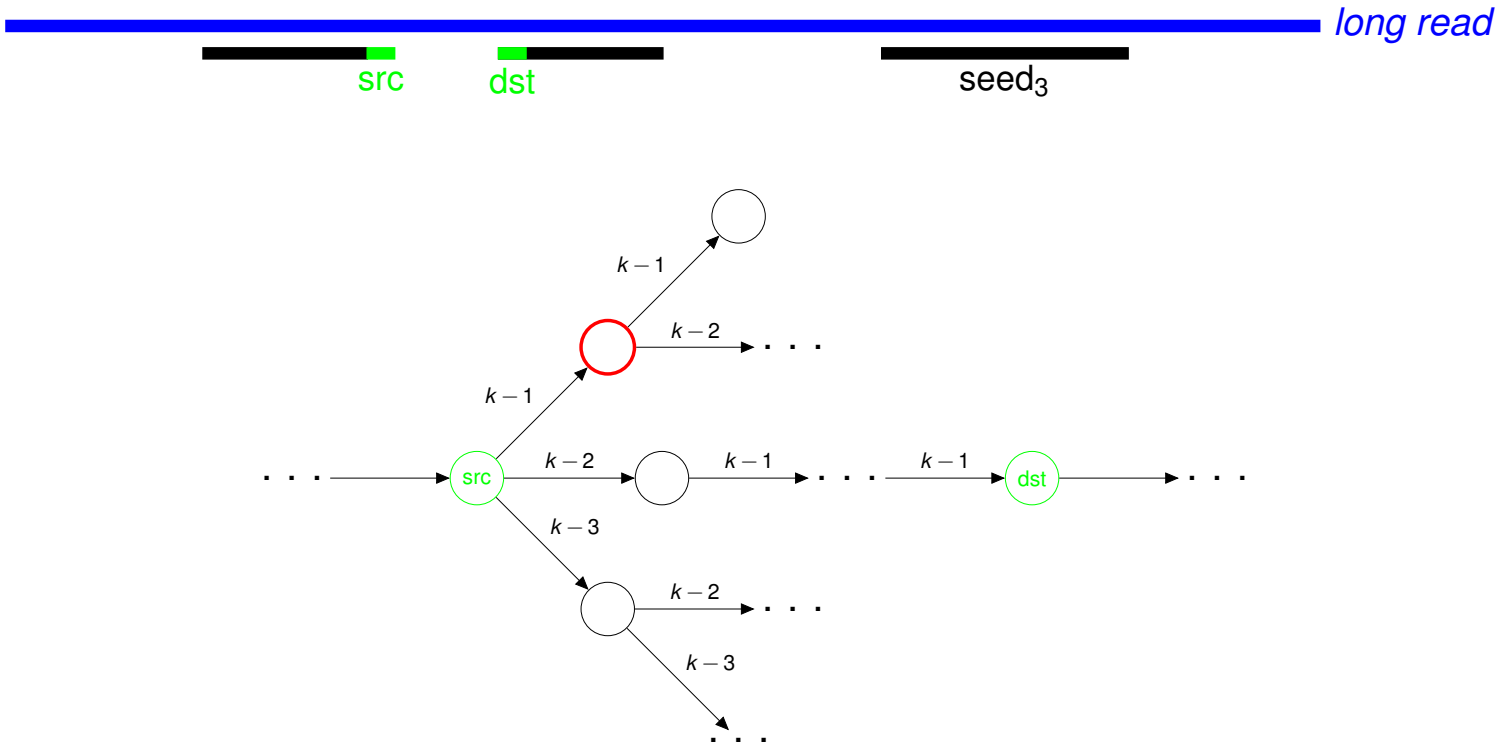
Step 4: Seeds linking



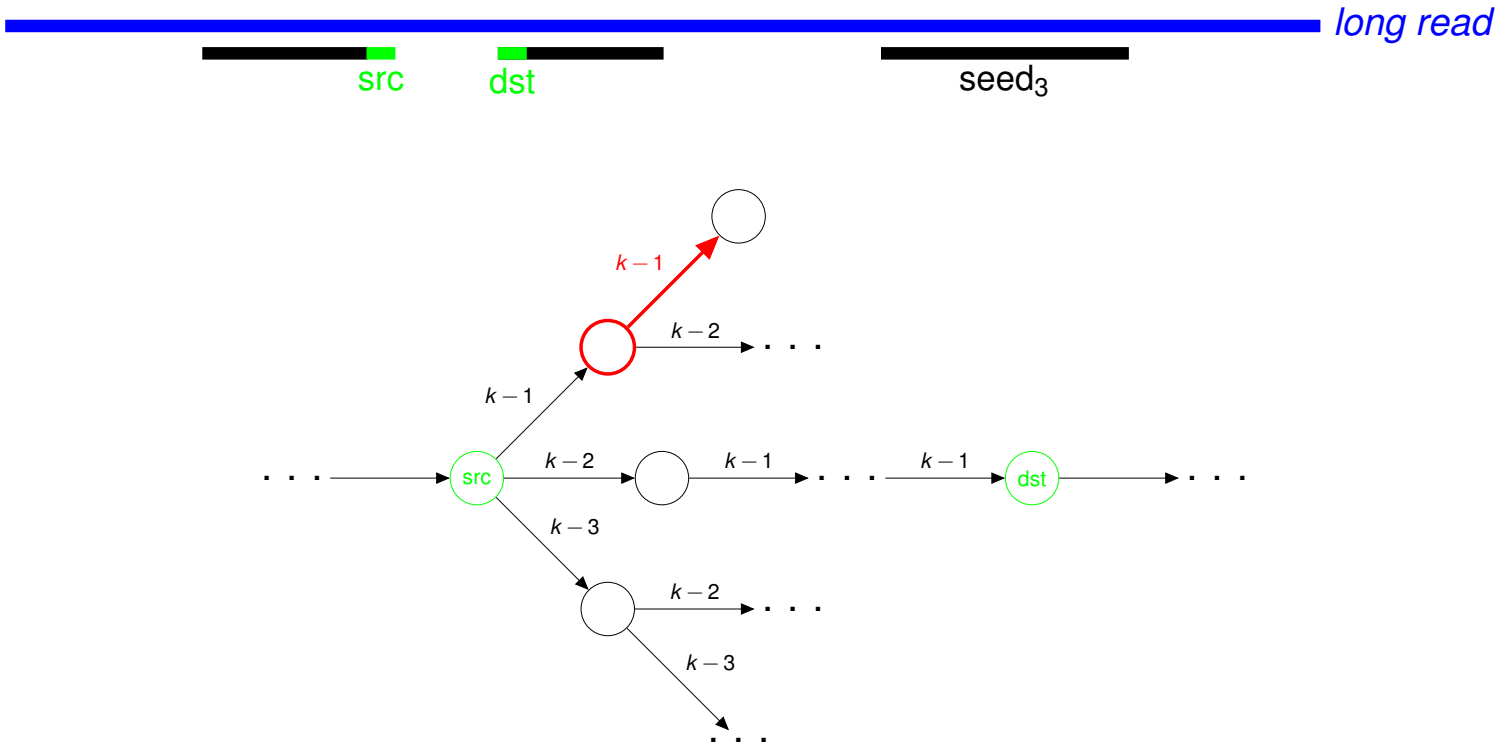
Step 4: Seeds linking



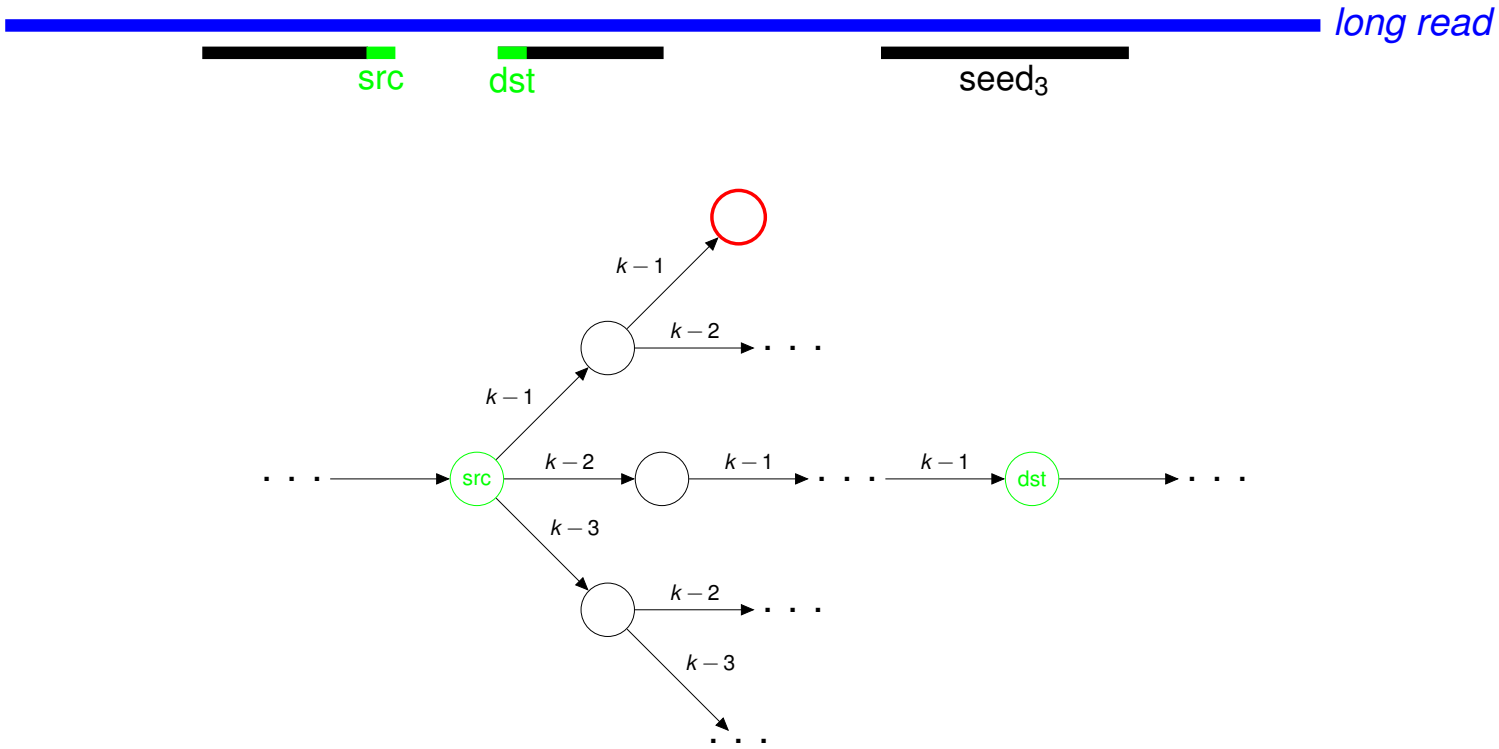
Step 4: Seeds linking



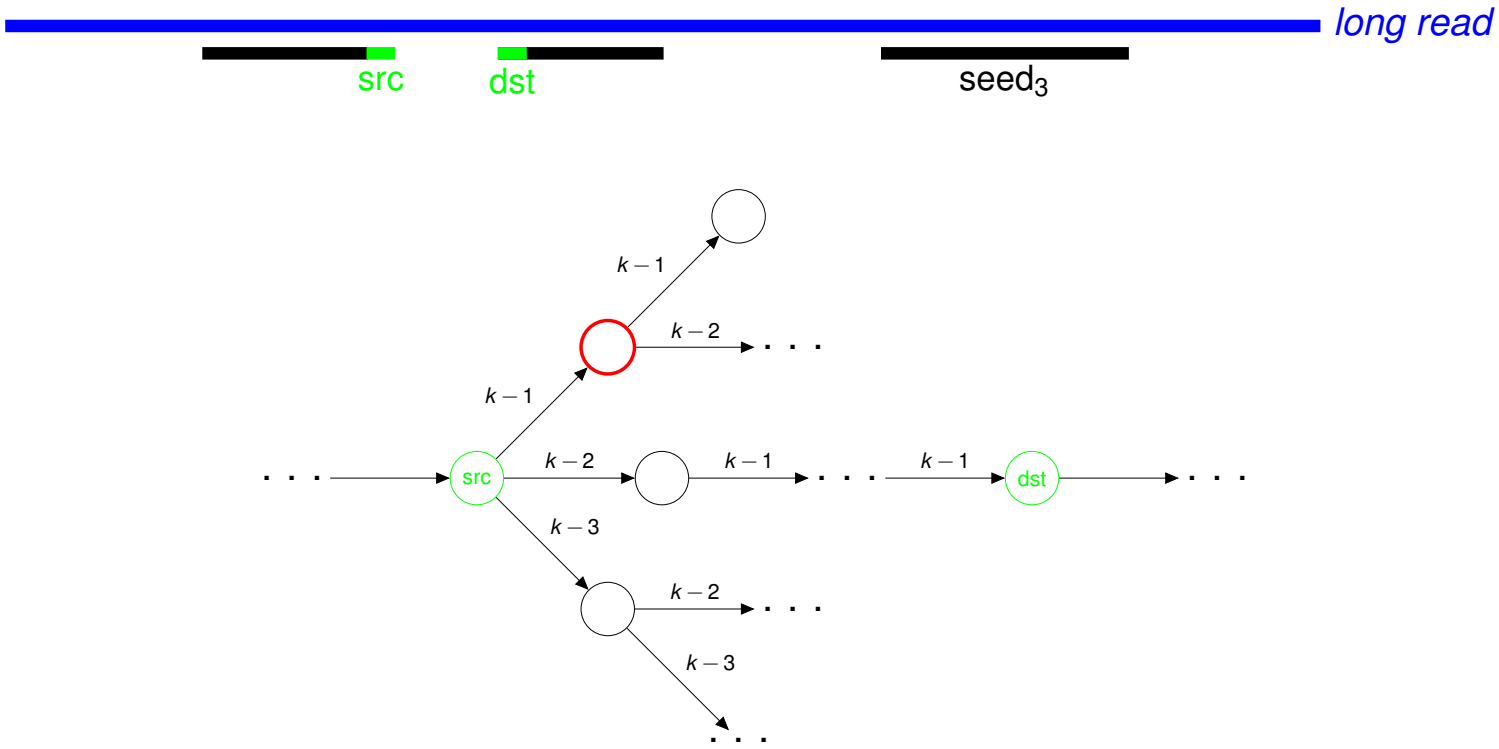
Step 4: Seeds linking



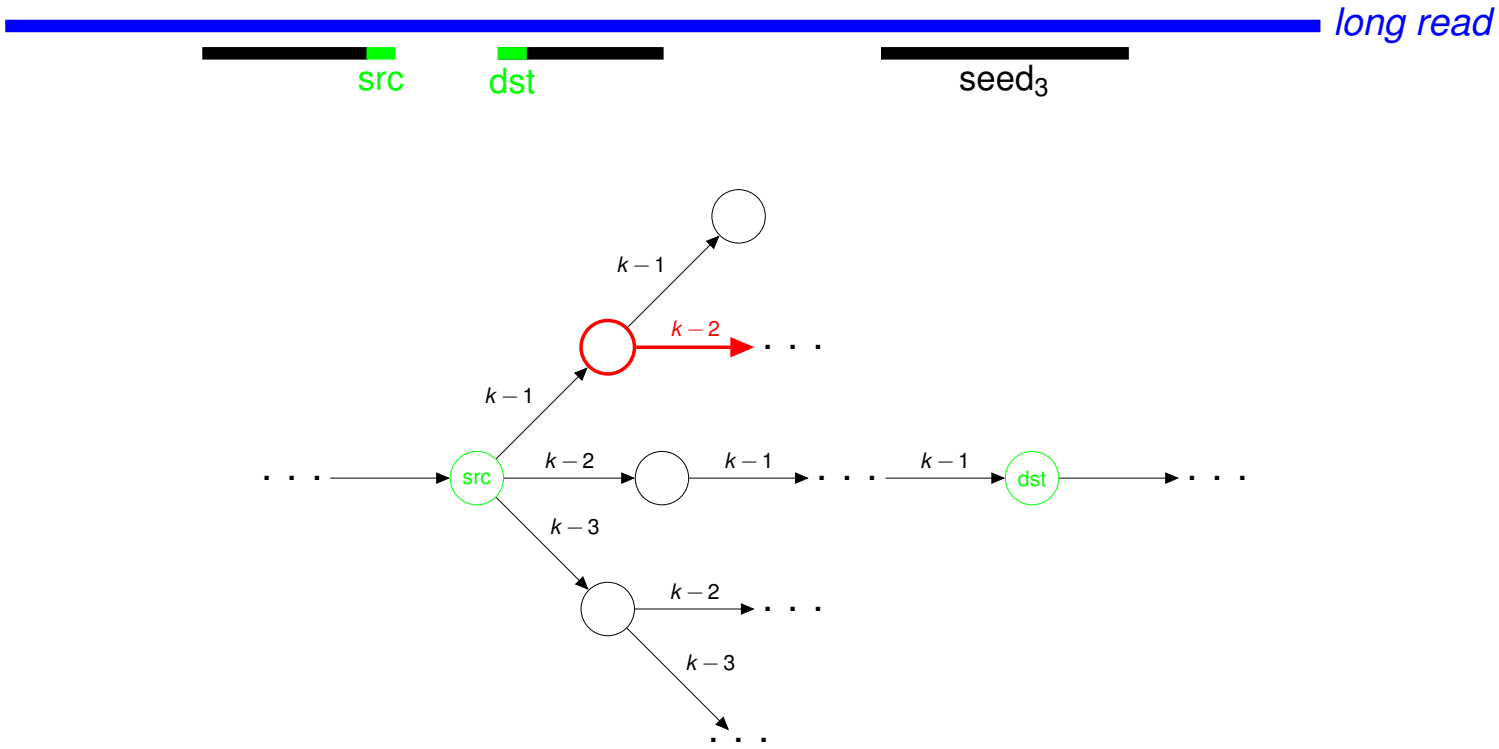
Step 4: Seeds linking



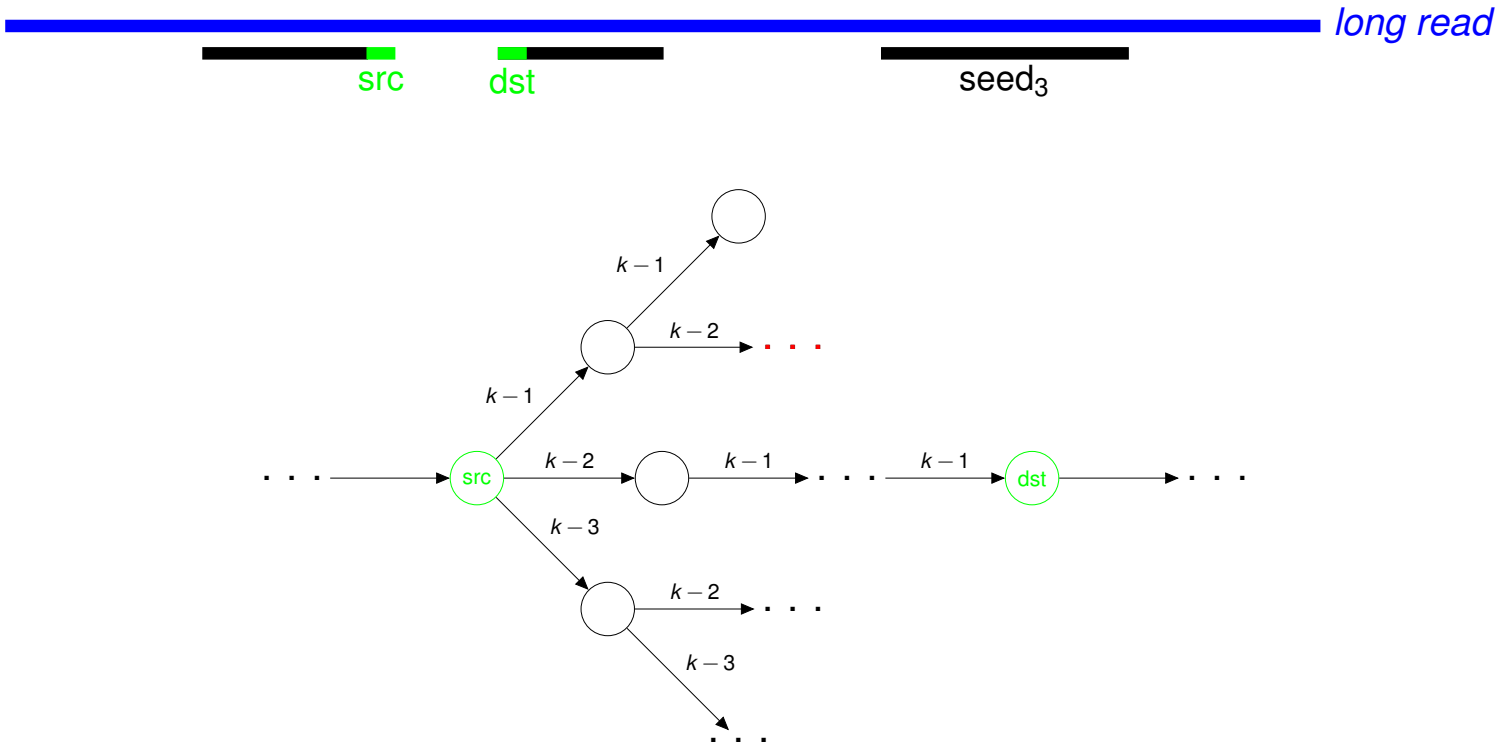
Step 4: Seeds linking



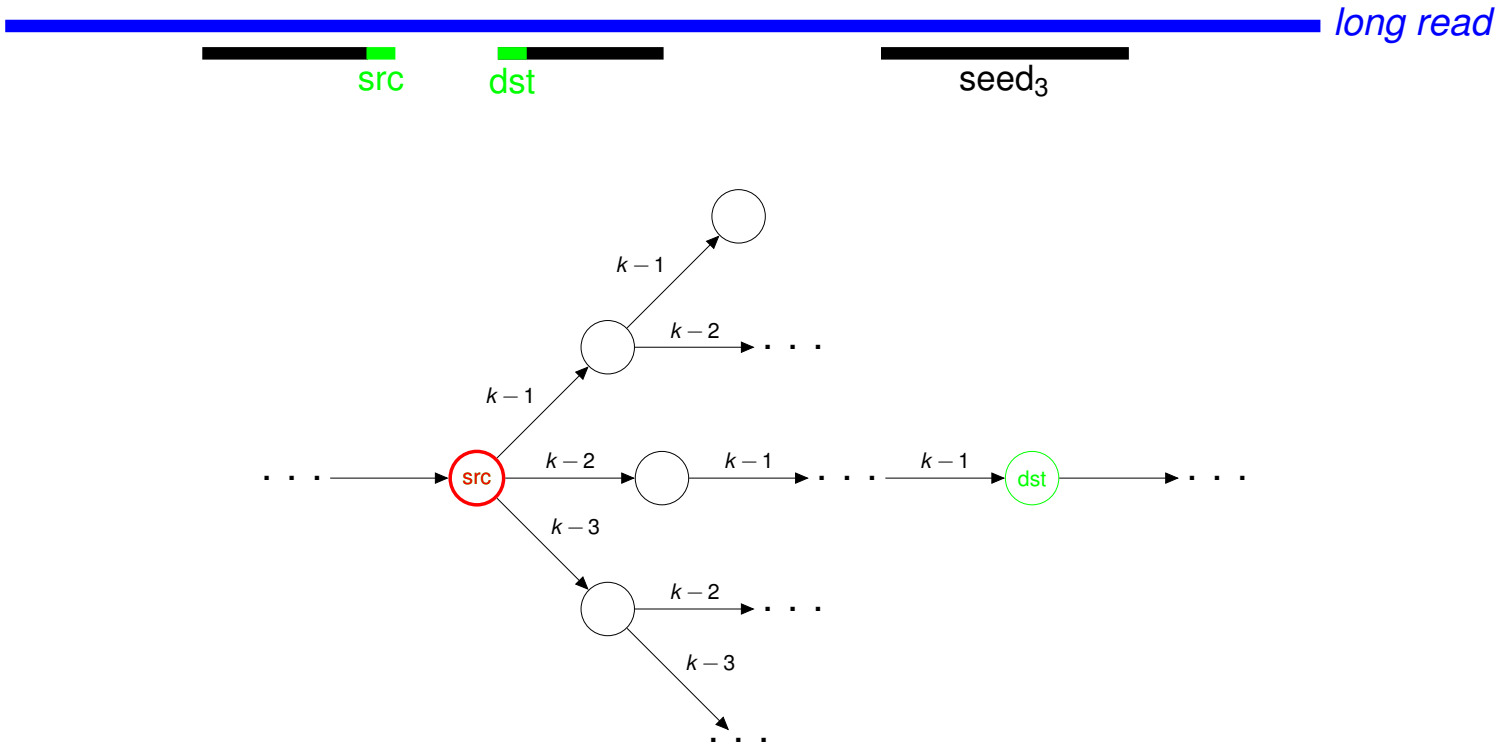
Step 4: Seeds linking



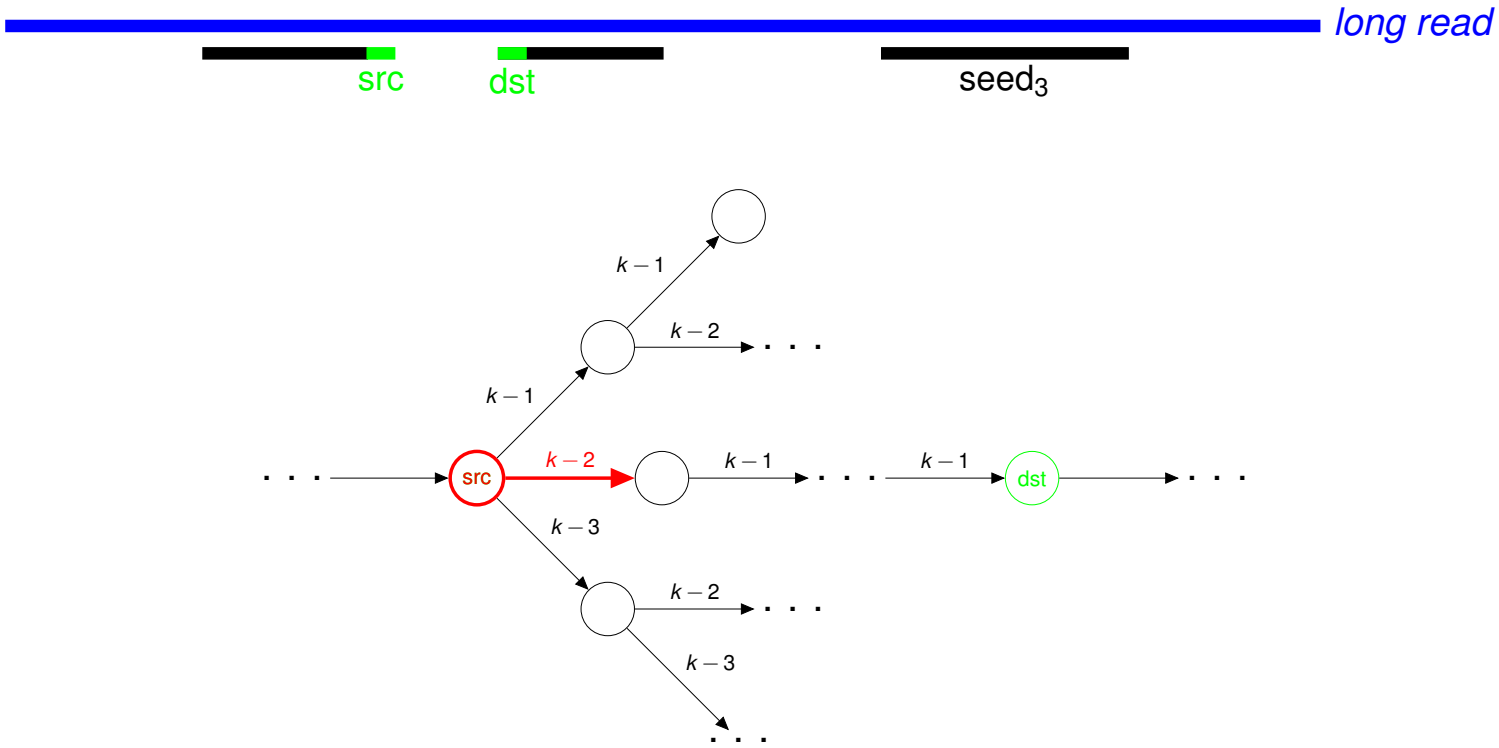
Step 4: Seeds linking



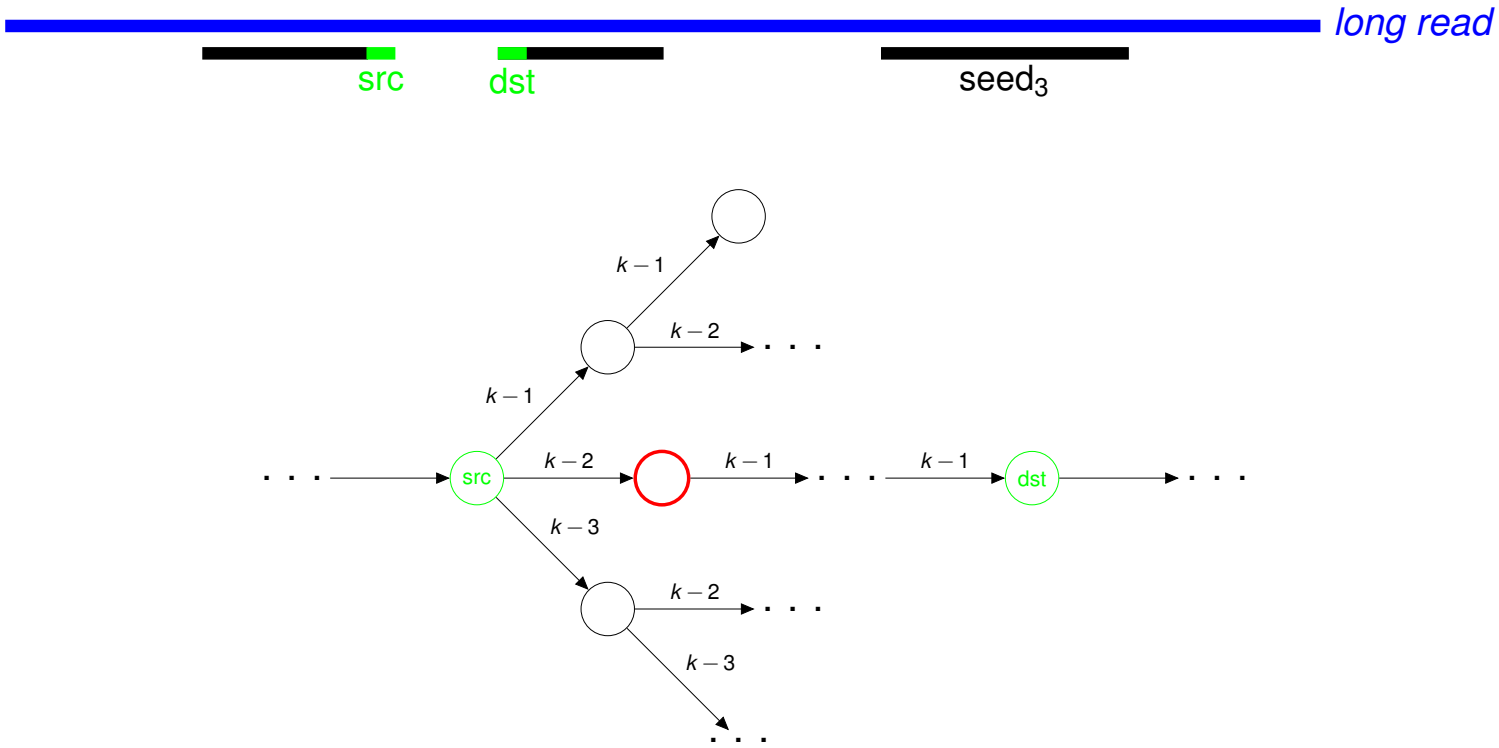
Step 4: Seeds linking



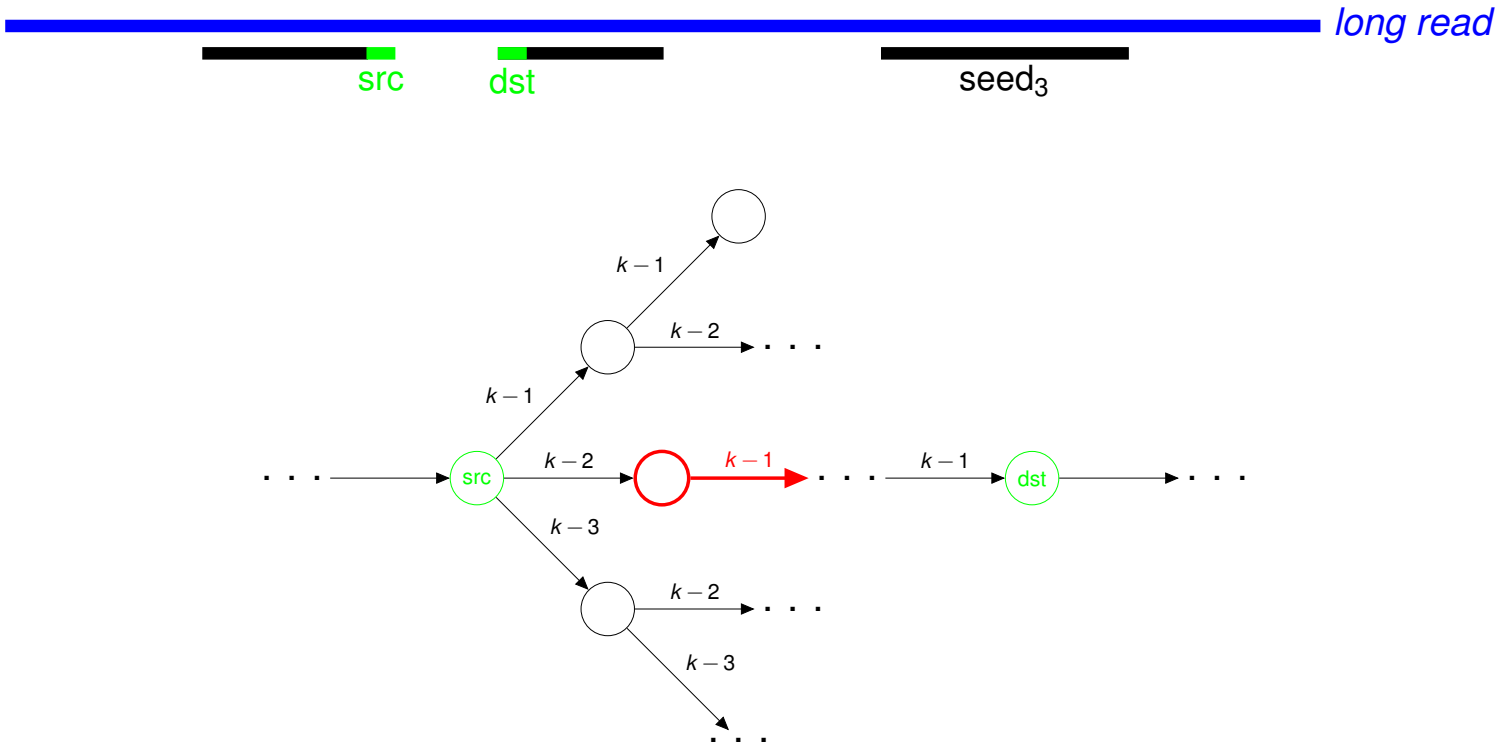
Step 4: Seeds linking



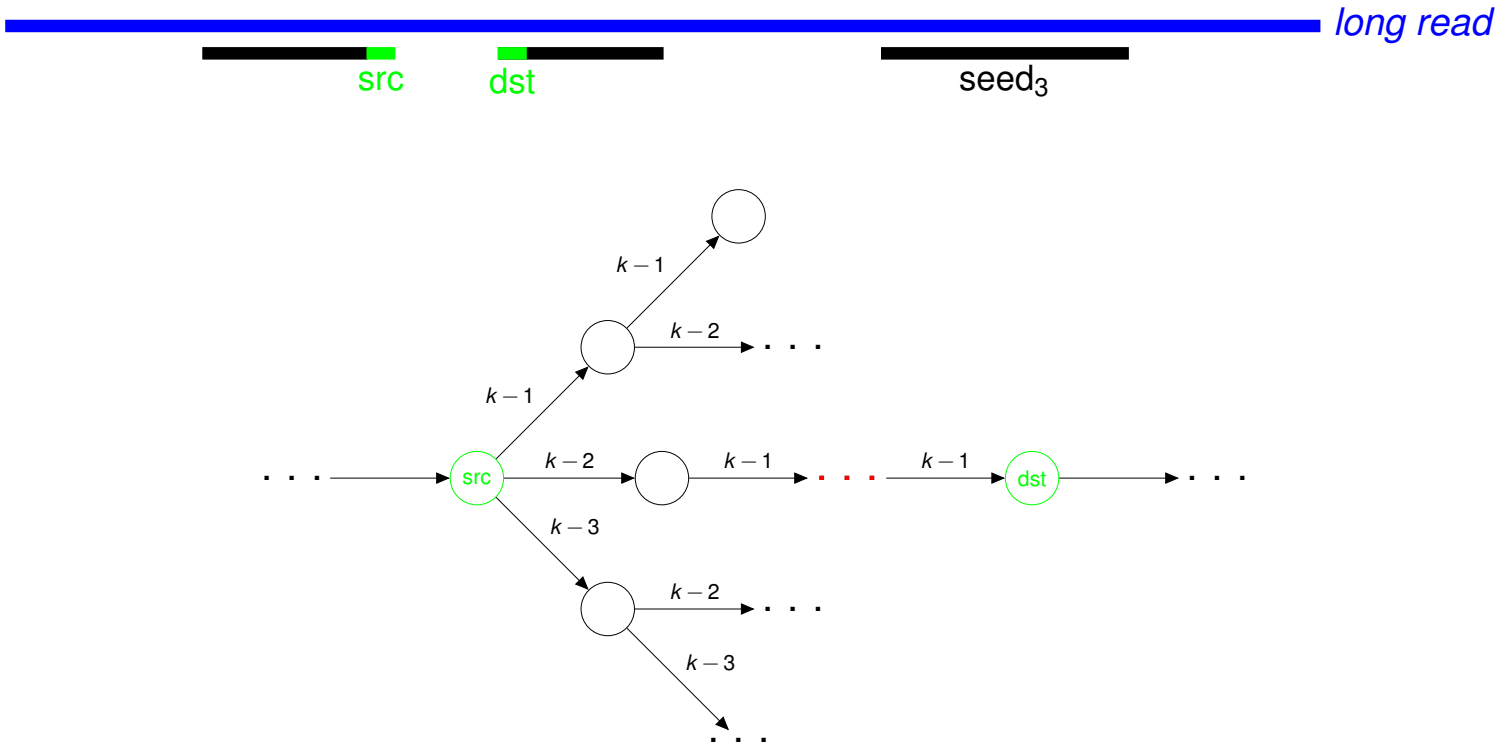
Step 4: Seeds linking



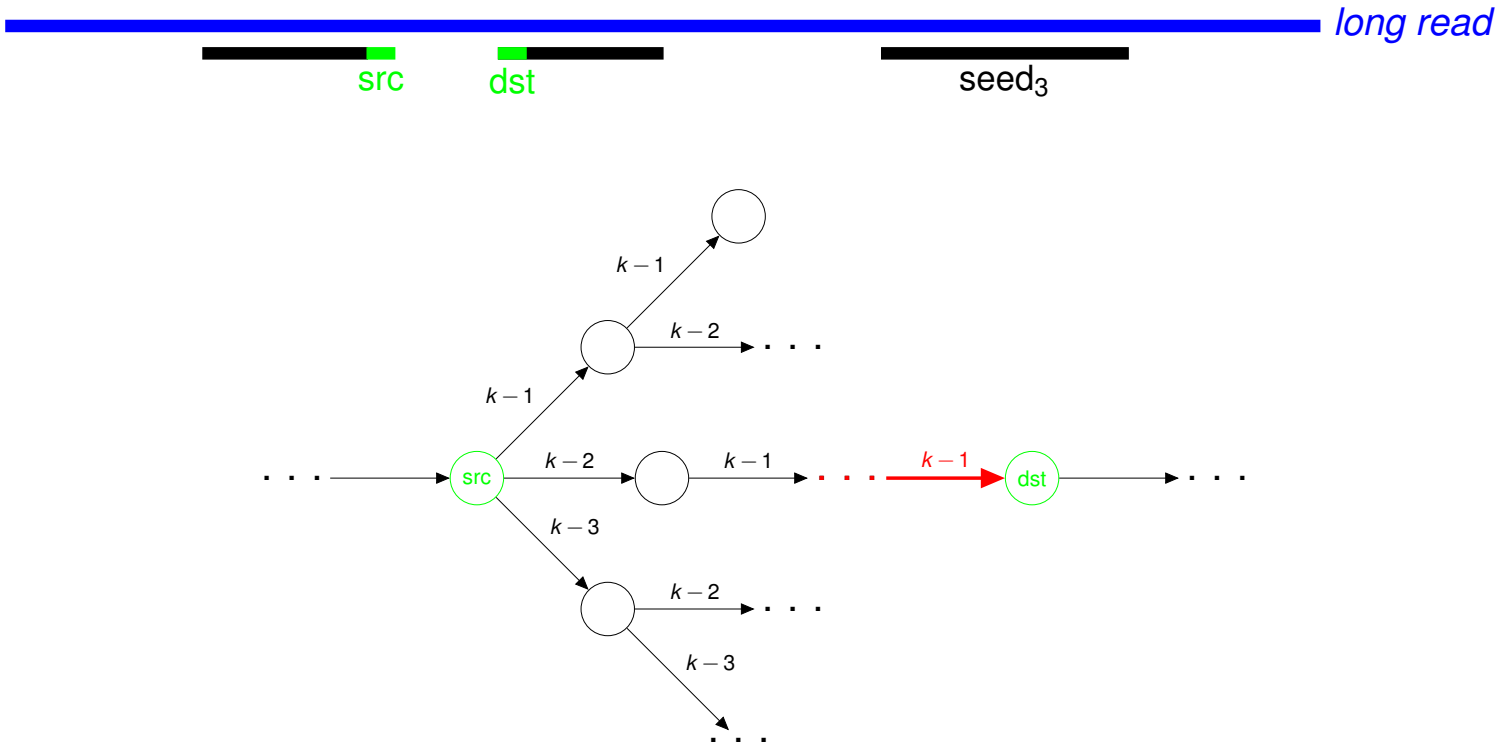
Step 4: Seeds linking



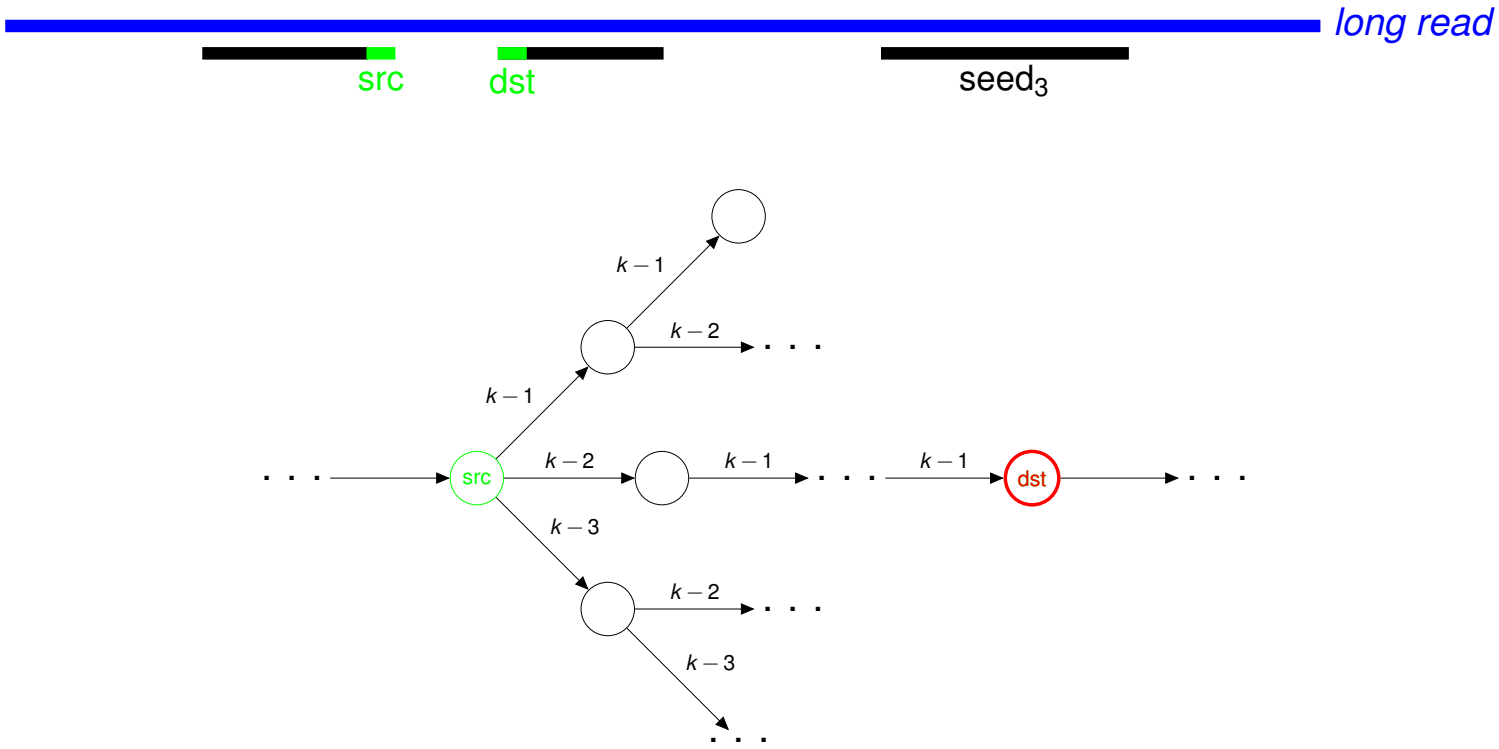
Step 4: Seeds linking



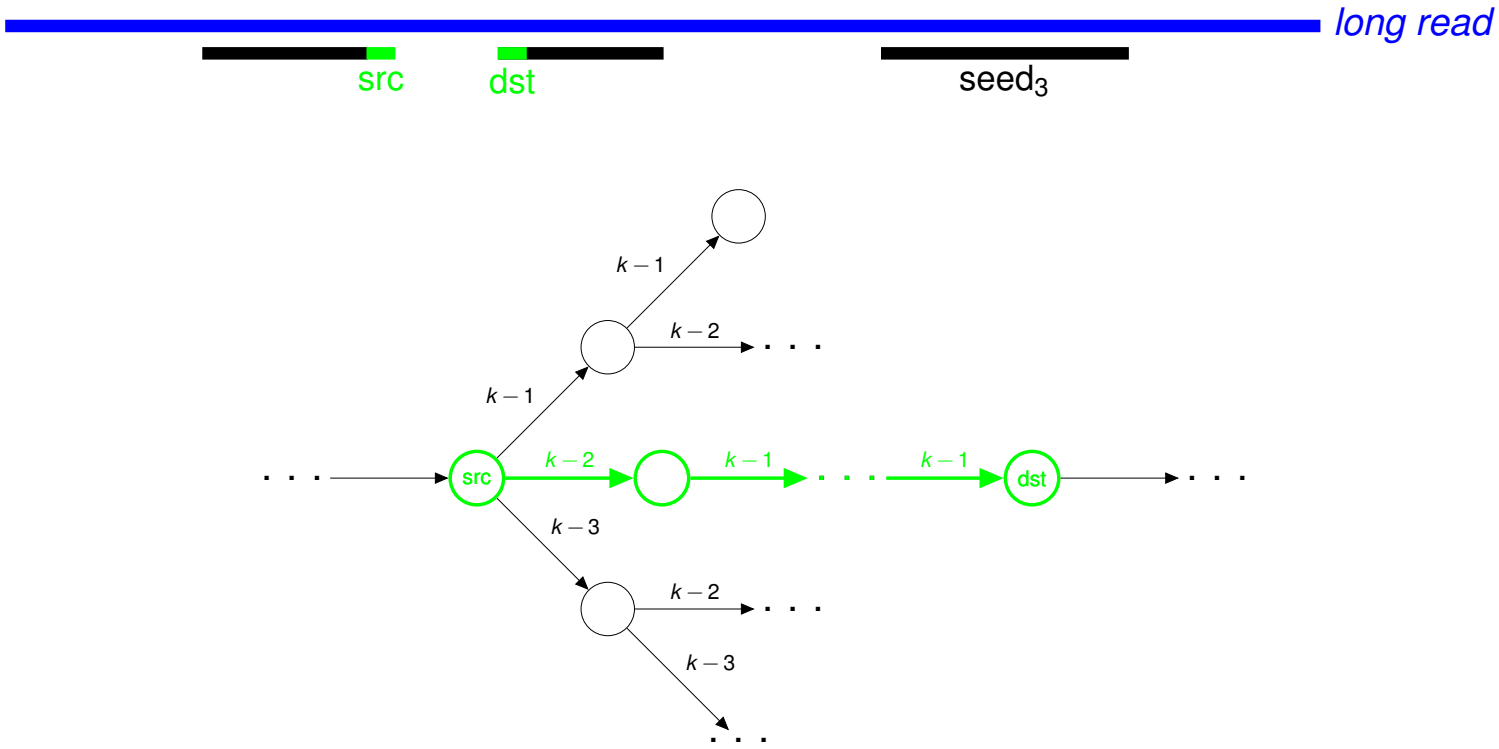
Step 4: Seeds linking



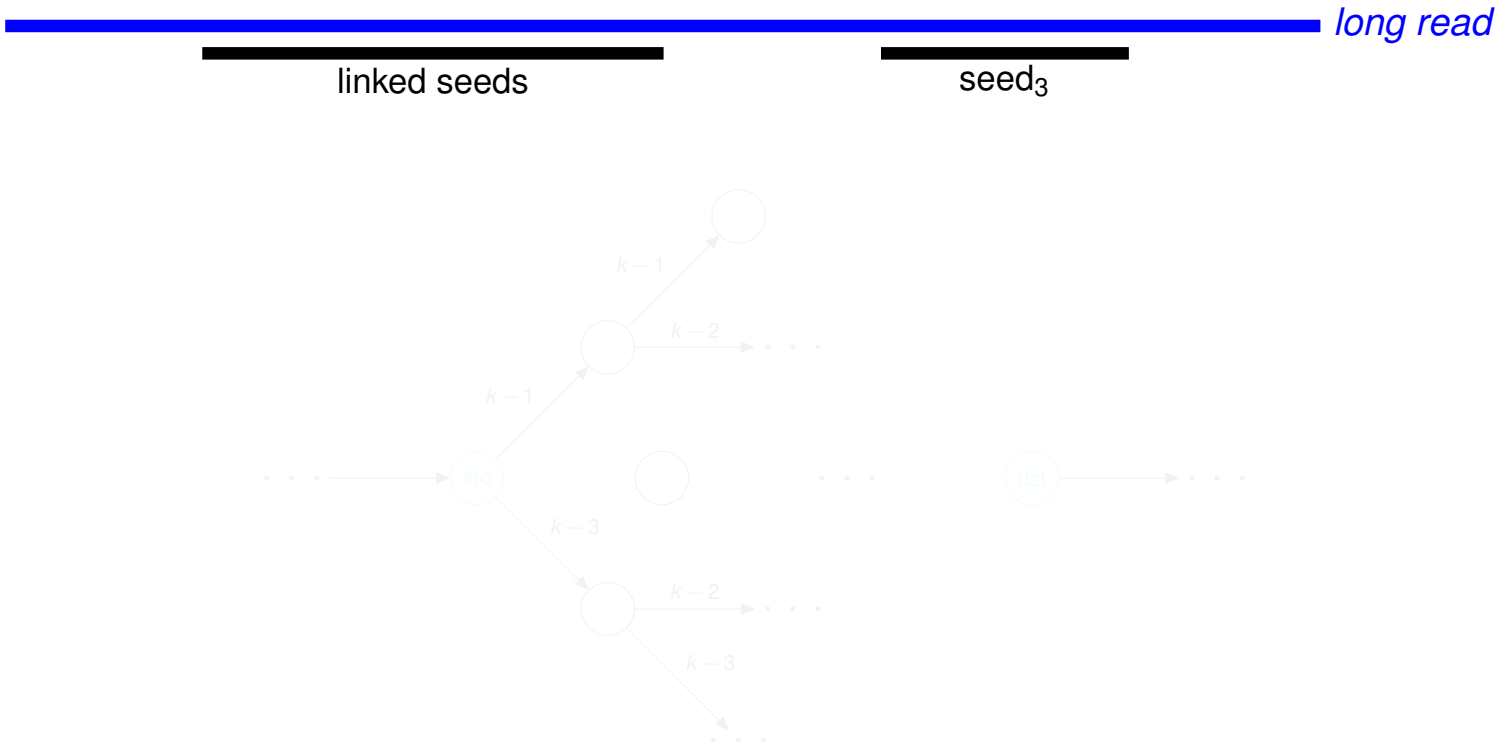
Step 4: Seeds linking



Step 4: Seeds linking

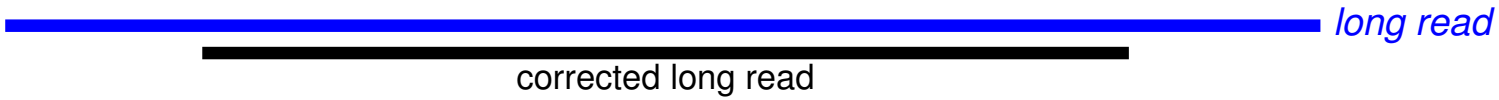


Step 4: Seeds linking



Step 4: Seeds linking

Step 4: Seeds linking



Step 5: Tips extension

- Seeds don't always map right at the beginning or until the end of the long read
- Once all the seeds have been linked, HG-CoLoR keeps on traversing the graph
- The traversal stops when the borders of the long read or a branching path are reached

Step 5: Tips extension

- Seeds don't always map right at the beginning or until the end of the long read
- Once all the seeds have been linked, HG-CoLoR keeps on traversing the graph
- The traversal stops when the borders of the long read or a branching path are reached

Step 5: Tips extension

- Seeds don't always map right at the beginning or until the end of the long read
- Once all the seeds have been linked, HG-CoLoR keeps on traversing the graph
- The traversal stops when the borders of the long read or a branching path are reached

Remark

- Some seeds might be impossible to link together
- $\Rightarrow$ Production of a corrected long read fragmented in multiple parts

Remark

- Some seeds might be impossible to link together
- $\Rightarrow$ Production of a corrected long read fragmented in multiple parts

Datasets

HG-CoLoR was compared to NaS, and two other state-of-the-art long read hybrid correction methods: CoLoRMap [Haghshenas et al., 2016] and Jabba [Miclotte et al., 2016]

The different tools were compared on the following datasets

Dataset	Reference genome			Oxford Nanopore data			Illumina data		
	Name	Strain	Genome size	# Reads	Average length	Coverage	# Reads	Read length	Coverage
E. coli	*E. coli*	K-12 substr. MG1655	4.6 Mbp	22,270	5,999	28x	775,500	300	50x
Yeast	*S. cerevisae*	W303	12.4 Mbp	205,923	5,698	31x	2,500,000	250	50x

Alignment-based comparison

Dataset	Method	# Reads	Average length	Average identity	Genome coverage	Runtime
E. coli	Original	22,270	5,999	79.46%	100%	N/A
	CoLoRMap	22,270	6,219	89.02%	100%	8h26min
	Jabba	22,065	5,794	99.81%	99.41%	12min56
	NaS	21,818	7,926	99.86%	100%	3 days
	HG-CoLoR	22,549	5,897	99.59%	100%	3h
Yeast	Original	205,923	5,698	55.49%	99.90%	N/A
	CoLoRMap	205,923	5,737	39.93%	99.40%	37h36min
	Jabba	36,958	6,613	99.55%	93.21%	44min05
	NaS	71,793	5,938	99.59%	98.70%	$>$ 16 days
	HG-CoLoR	71,518	6,604	99.17%	98.39%	22h

Alignment-based comparison

Dataset	Method	# Reads	Average length	Average identity	Genome coverage	Runtime
E. coli	Original	22,270	5,999	79.46%	100%	N/A
	CoLoRMap	**22,270**	**6,219**	**89.02%**	**100%**	**8h26min**
	Jabba	22,065	5,794	99.81%	99.41%	12min56
	NaS	21,818	7,926	99.86%	100%	3 days
	HG-CoLoR	22,549	5,897	99.59%	100%	3h
Yeast	Original	205,923	5,698	55.49%	99.90%	N/A
	CoLoRMap	**205,923**	**5,737**	**39.93%**	**99.40%**	**37h36min**
	Jabba	36,958	6,613	99.55%	93.21%	44min05
	NaS	71,793	5,938	99.59%	98.70%	$>$ 16 days
	HG-CoLoR	71,518	6,604	99.17%	98.39%	22h

Alignment-based comparison

Dataset	Method	# Reads	Average length	Average identity	Genome coverage	Runtime
E. coli	Original	22,270	5,999	79.46%	100%	N/A
	CoLoRMap	22,270	6,219	89.02%	100%	8h26min
	Jabba	**22,065**	**5,794**	**99.81%**	**99.41%**	**12min56**
	NaS	21,818	7,926	99.86%	100%	3 days
	HG-CoLoR	22,549	5,897	99.59%	100%	3h
Yeast	Original	205,923	5,698	55.49%	99.90%	N/A
	CoLoRMap	205,923	5,737	39.93%	99.40%	37h36min
	Jabba	**36,958**	**6,613**	**99.55%**	**93.21%**	**44min05**
	NaS	71,793	5,938	99.59%	98.70%	$>$ 16 days
	HG-CoLoR	71,518	6,604	99.17%	98.39%	22h

Alignment-based comparison

Dataset	Method	# Reads	Average length	Average identity	Genome coverage	Runtime
E. coli	Original	22,270	5,999	79.46%	100%	N/A
	CoLoRMap	22,270	6,219	89.02%	100%	8h26min
	Jabba	22,065	5,794	99.81%	99.41%	12min56
	NaS	**21,818**	**7,926**	**99.86%**	**100%**	**3 days**
	HG-CoLoR	22,549	5,897	99.59%	100%	3h
Yeast	Original	205,923	5,698	55.49%	99.90%	N/A
	CoLoRMap	205,923	5,737	39.93%	99.40%	37h36min
	Jabba	36,958	6,613	99.55%	93.21%	44min05
	NaS	**71,793**	**5,938**	**99.59%**	**98.70%**	**> 16 days**
	HG-CoLoR	71,518	6,604	99.17%	98.39%	22h

Alignment-based comparison

Dataset	Method	# Reads	Average length	Average identity	Genome coverage	Runtime
E. coli	Original	22,270	5,999	79.46%	100%	N/A
	CoLoRMap	22,270	6,219	89.02%	100%	8h26min
	Jabba	22,065	5,794	99.81%	99.41%	12min56
	NaS	21,818	7,926	99.86%	100%	3 days
	HG-CoLoR	**22,549**	**5,897**	**99.59%**	**100%**	**3h**
Yeast	Original	205,923	5,698	55.49%	99.90%	N/A
	CoLoRMap	205,923	5,737	39.93%	99.40%	37h36min
	Jabba	36,958	6,613	99.55%	93.21%	44min05
	NaS	71,793	5,938	99.59%	98.70%	$>$ 16 days
	HG-CoLoR	**71,518**	**6,604**	**99.17%**	**98.39%**	**22h**

Assembly-based comparison

Dataset	Method	Coverage	# Expected contigs	# Obtained contigs	Genome coverage	Identity
E. coli	CoLoRMap	28x	1	29	97,74%	99.81%
	Jabba	28x	1	41	95.76%	99.92%
	NaS	37x	1	1	99.90%	99.99%
	HG-CoLoR	29x	1	2	99.95%	99.95%
Yeast	CoLoRMap	14x	30	-	-	-
	Jabba	21x	30	134	70.52%	99.83%
	NaS	35x	30	123	97.44%	99.77%
	HG-CoLoR	39x	30	108	92.19%	99.61%

Assembly-based comparison

Dataset	Method	Coverage	# Expected contigs	# Obtained contigs	Genome coverage	Identity
E. coli	**CoLoRMap**	**28x**	**1**	**29**	**97,74%**	**99.81%**
	Jabba	28x	1	41	95.76%	99.92%
	NaS	37x	1	1	99.90%	99.99%
	HG-CoLoR	29x	1	2	99.95%	99.95%
Yeast	**CoLoRMap**	**14x**	**30**	-	-	-
	Jabba	21x	30	134	70.52%	99.83%
	NaS	35x	30	123	97.44%	99.77%
	HG-CoLoR	39x	30	108	92.19%	99.61%

Assembly-based comparison

Dataset	Method	Coverage	# Expected contigs	# Obtained contigs	Genome coverage	Identity
E. coli	CoLoRMap	28x	1	29	97,74%	99.81%
	Jabba	**28x**	**1**	**41**	**95.76%**	**99.92%**
	NaS	37x	1	1	99.90%	99.99%
	HG-CoLoR	29x	1	2	99.95%	99.95%
Yeast	CoLoRMap	14x	30	-	-	-
	Jabba	**21x**	**30**	**134**	**70.52%**	**99.83%**
	NaS	35x	30	123	97.44%	99.77%
	HG-CoLoR	39x	30	108	92.19%	99.61%

Assembly-based comparison

Dataset	Method	Coverage	# Expected contigs	# Obtained contigs	Genome coverage	Identity
E. coli	CoLoRMap	28x	1	29	97,74%	99.81%
	Jabba	28x	1	41	95.76%	99.92%
	NaS	**37x**	**1**	**1**	**99.90%**	**99.99%**
	HG-CoLoR	29x	1	2	99.95%	99.95%
Yeast	CoLoRMap	14x	30	-	-	-
	Jabba	21x	30	134	70.52%	99.83%
	NaS	**35x**	**30**	**123**	**97.44%**	**99.77%**
	HG-CoLoR	39x	30	108	92.19%	99.61%

Assembly-based comparison

Dataset	Method	Coverage	# Expected contigs	# Obtained contigs	Genome coverage	Identity
E. coli	CoLoRMap	28x	1	29	97,74%	99.81%
	Jabba	28x	1	41	95.76%	99.92%
	NaS	37x	1	1	99.90%	99.99%
	HG-CoLoR	**29x**	**1**	**2**	**99.95%**	**99.95%**
Yeast	CoLoRMap	14x	30	-	-	-
	Jabba	21x	30	134	70.52%	99.83%
	NaS	35x	30	123	97.44%	99.77%
	HG-CoLoR	**39x**	**30**	**108**	**92.19%**	**99.61%**

Conclusion

- We introduced a new graph structure and proved its usefulness
- We developed a new hybrid long read error correction method
- We showed that this new method provides the best trade off between runtime, accuracy and genome coverage, when compared to state-of-the-art methods
- HG-CoLoR is available from: https://github.com/pierre-morisse/HG-CoLoR

Conclusion

- We introduced a new graph structure and proved its usefulness
- We developed a new hybrid long read error correction method
- We showed that this new method provides the best trade off between runtime, accuracy and genome coverage, when compared to state-of-the-art methods
- HG-CoLoR is available from: https://github.com/pierre-morisse/HG-CoLoR

Conclusion

- We introduced a new graph structure and proved its usefulness
- We developed a new hybrid long read error correction method
- We showed that this new method provides the best trade off between runtime, accuracy and genome coverage, when compared to state-of-the-art methods
- HG-CoLoR is available from:
  https://github.com/pierre-morisse/HG-CoLoR

Conclusion

- We introduced a new graph structure and proved its usefulness
- We developed a new hybrid long read error correction method
- We showed that this new method provides the best trade off between runtime, accuracy and genome coverage, when compared to state-of-the-art methods
- HG-CoLoR is available from:
  `https://github.com/pierre-morisse/HG-CoLoR`

Future work

- Run HG-CoLoR on larger genomes
- Filter out weak *k*-mers after the short reads correction step
- Build a proper assembly tool from the hybrid graph structure
- Adapt HG-CoLoR to self-correction

Future work

- Run HG-CoLoR on larger genomes
- Filter out weak *k*-mers after the short reads correction step
- Build a proper assembly tool from the hybrid graph structure
- Adapt HG-CoLoR to self-correction

Future work

- Run HG-CoLoR on larger genomes
- Filter out weak *k*-mers after the short reads correction step
- Build a proper assembly tool from the hybrid graph structure
- Adapt HG-CoLoR to self-correction

Future work

- Run HG-CoLoR on larger genomes
- Filter out weak *k*-mers after the short reads correction step
- Build a proper assembly tool from the hybrid graph structure
- Adapt HG-CoLoR to self-correction

References I

Haghshenas, E., Hach, F., Sahinalp, S. C., and Chauve, C. (2016).
CoLoRMap: Correcting Long Reads by Mapping short reads.
Bioinformatics, 32(17):i545–i551.

Kowalski, T., Grabowski, S., and Deorowicz, S. (2015).
Indexing arbitrary-length k-mers in sequencing reads.
PLoS ONE, 10(7):1–14.

Madoui, M.-A., Engelen, S., Cruaud, C., Belser, C., Bertrand, L., Alberti, A., Lemainque, A., Wincker, P., and Aury, J.-M. (2015).
Genome assembly using Nanopore-guided long and error-free DNA reads.
BMC Genomics, 16:327.

References II

Miclotte, G., Heydari, M., Demeester, P., Rombauts, S., Van de Peer, Y., Audenaert, P., and Fostier, J. (2016).
Jabba: hybrid error correction for long sequencing reads.
Algorithms Mol Biol, 11:10.

Questions?

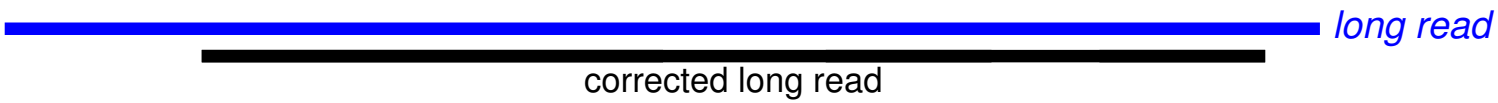
Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads



Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads

Fragmented corrected long reads

Hybrid graph traversal

Example

k-mers set

1: AAGCTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA
Index

Hybrid graph traversal

Example

k-mers set

1: **AAGCTT**
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA
Index

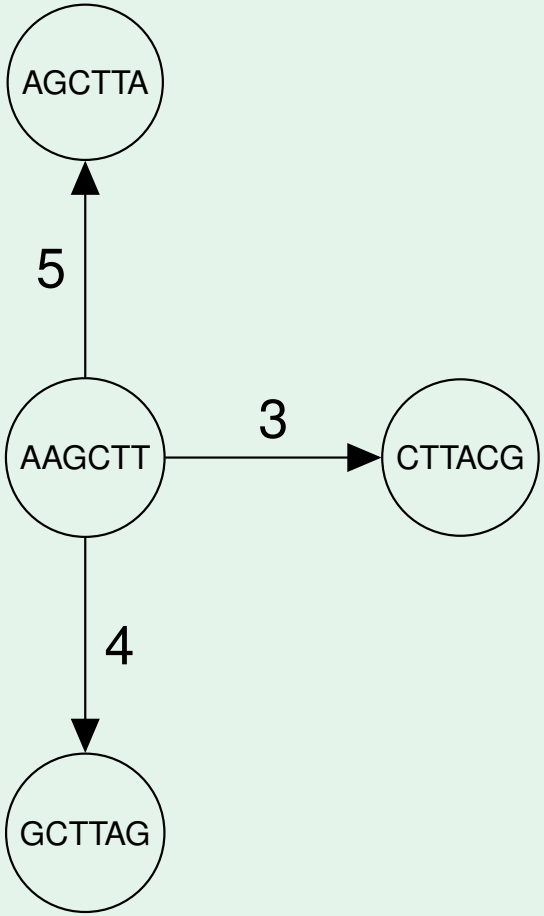
Hybrid graph traversal

Example

k-mers set

1: **AAGCTT**
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index



Hybrid graph traversal

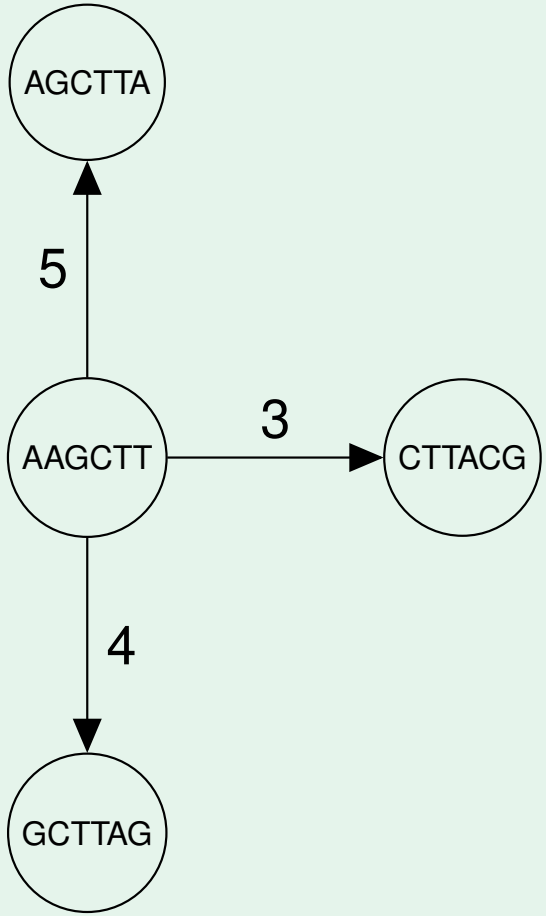
Example

k-mers set

1: **A**AGCTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

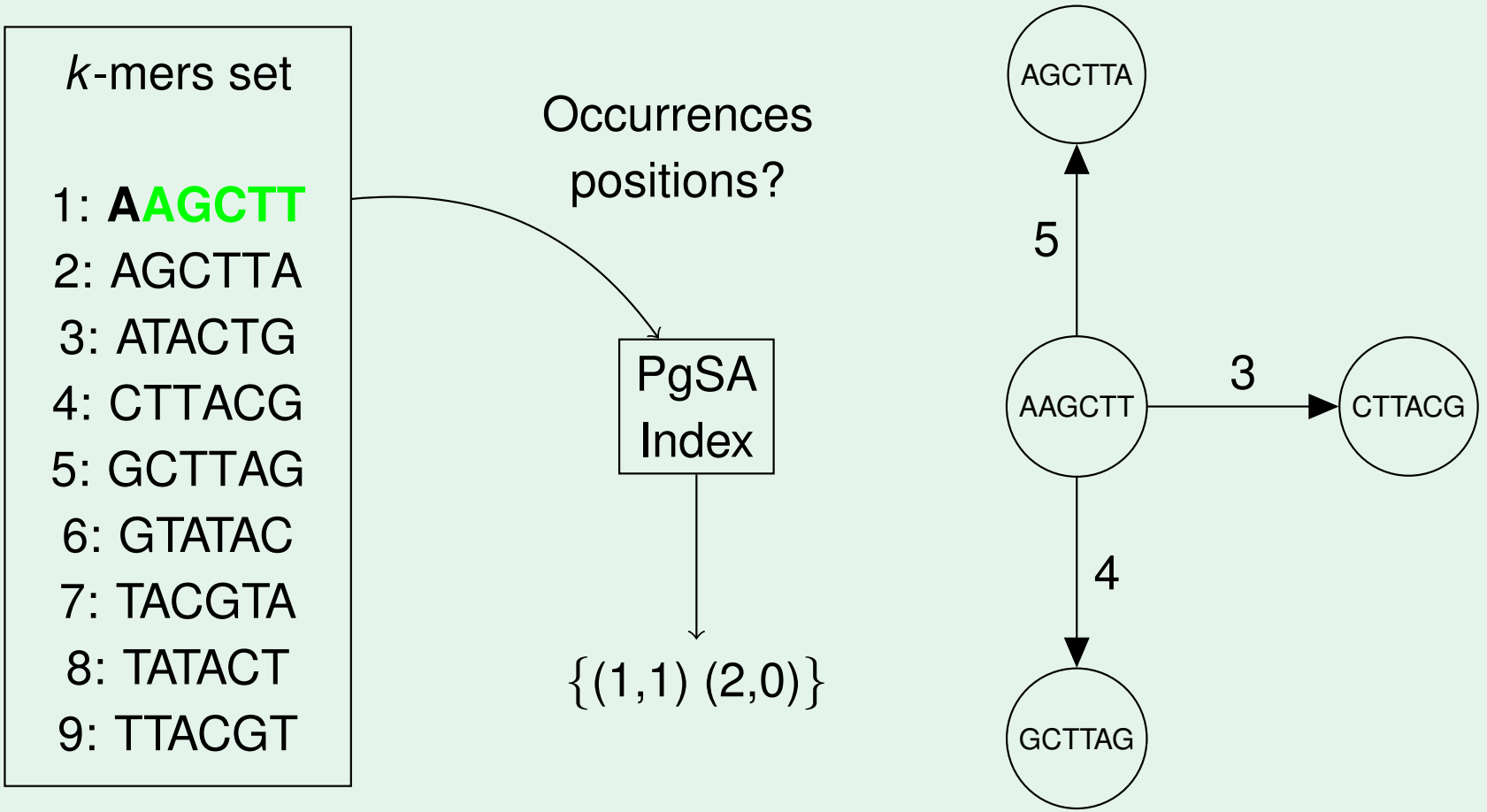
Occurrences positions?

PgSA Index



Hybrid graph traversal

Example



Hybrid graph traversal

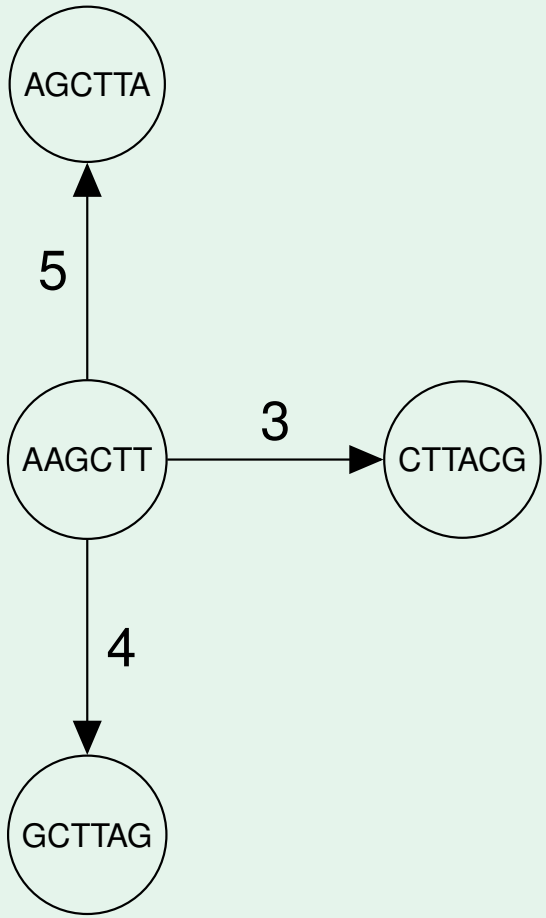
Example

k-mers set

1: **A**AGCTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index

{(1,1) (2,0)}



Hybrid graph traversal

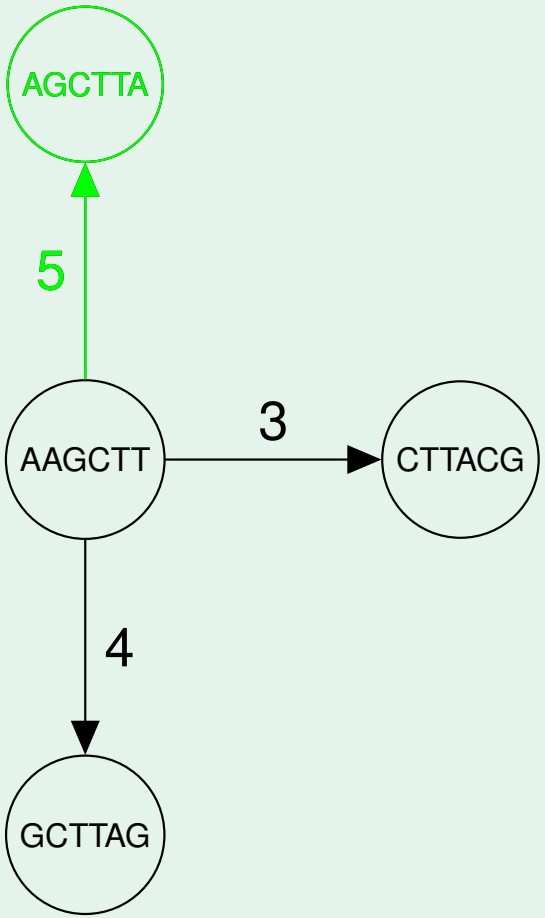
Example

k-mers set

1: **A**AGCTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

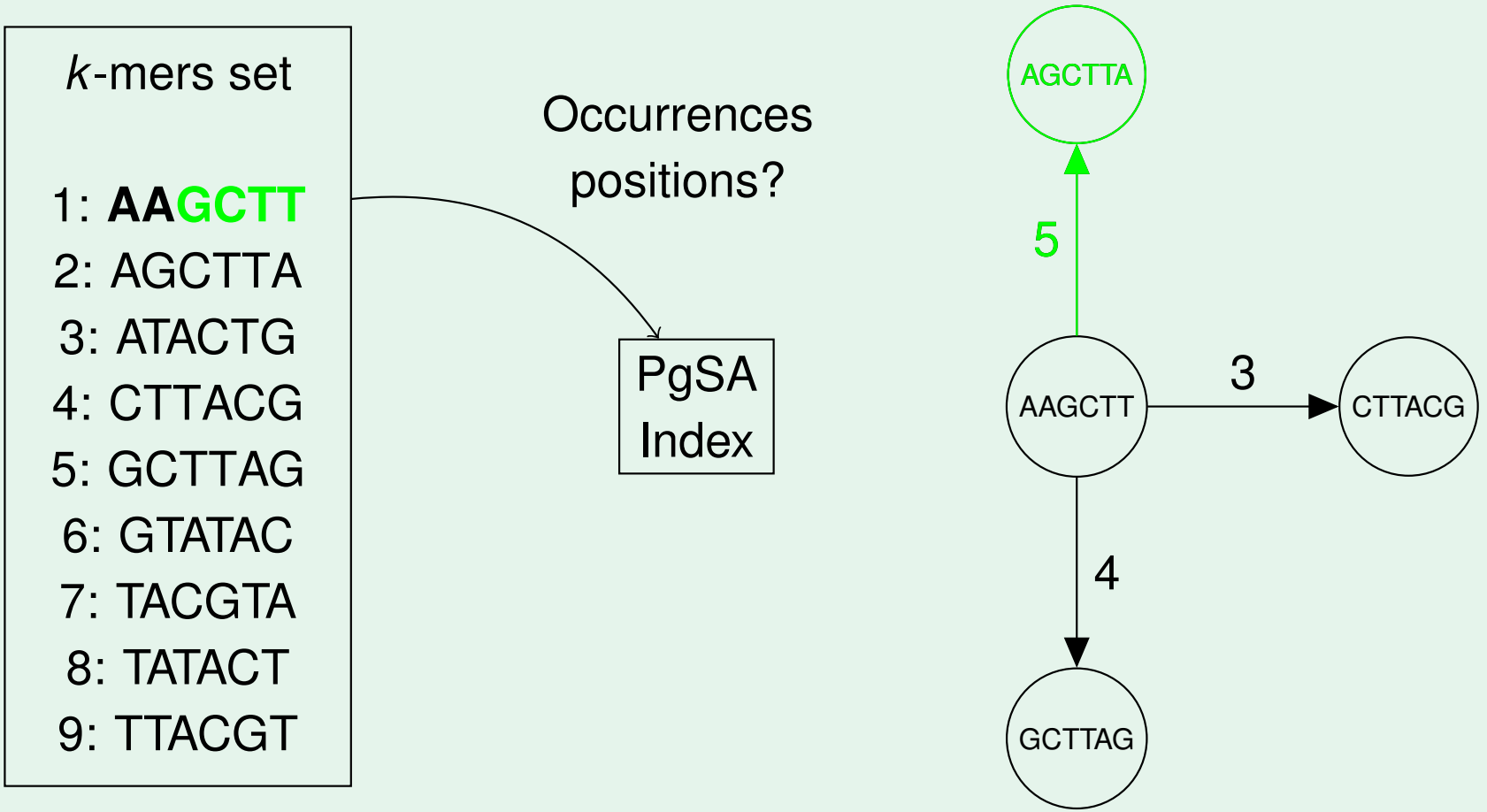
PgSA Index

{(1,1) (2,0)}



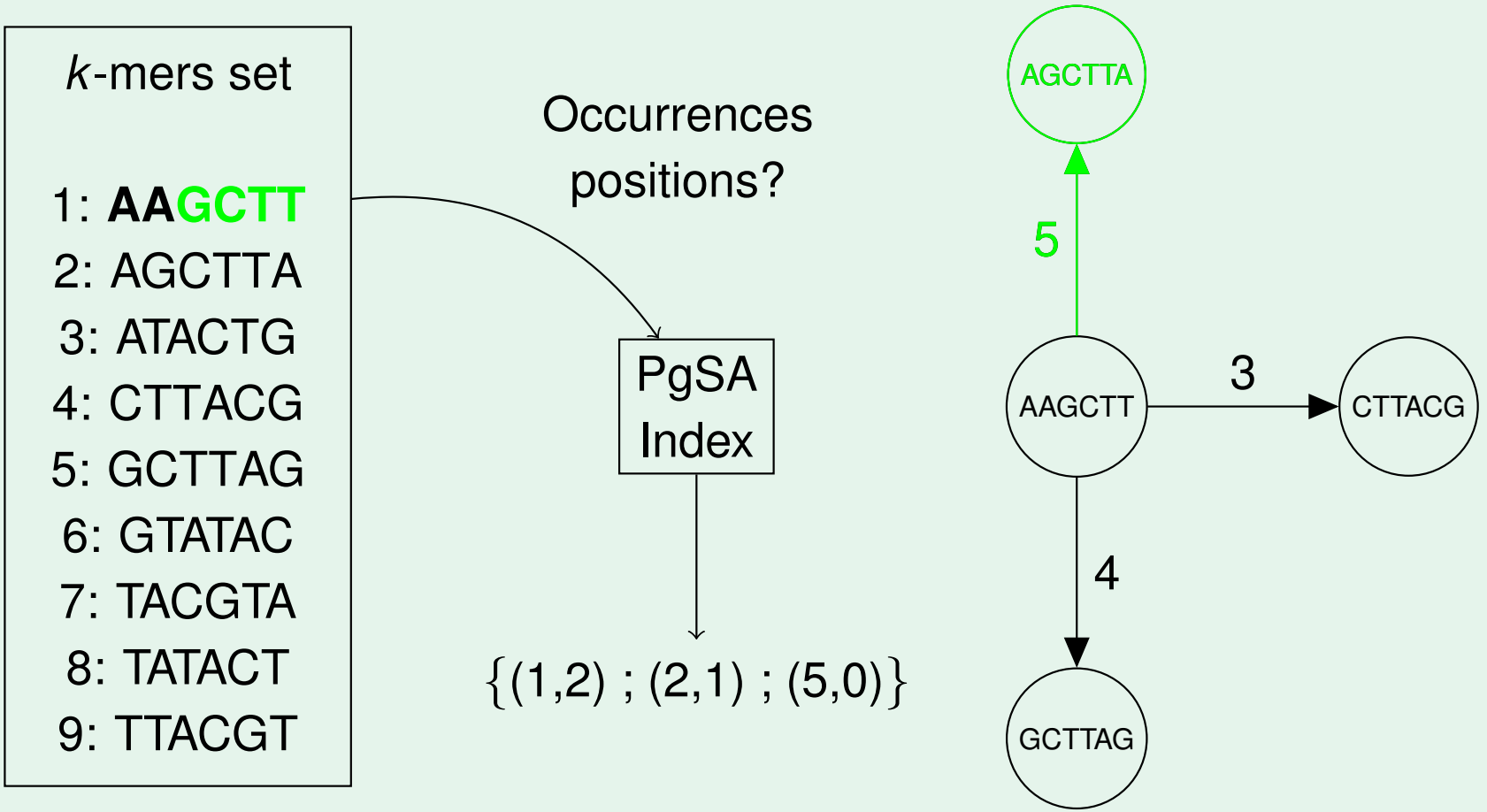
Hybrid graph traversal

Example



Hybrid graph traversal

Example



Hybrid graph traversal

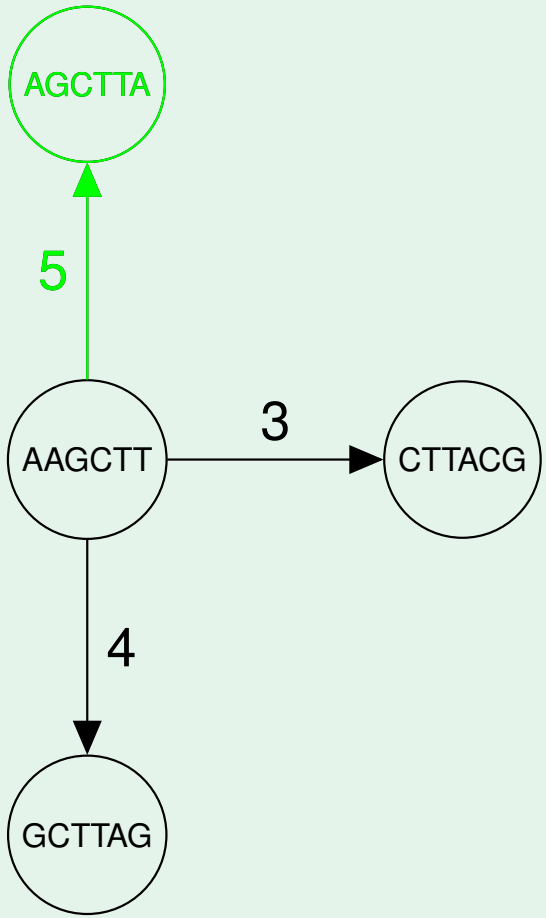
Example

k-mers set

1: **AA**GCTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index

{(1,2) ; (2,1) ; (5,0)}



Hybrid graph traversal

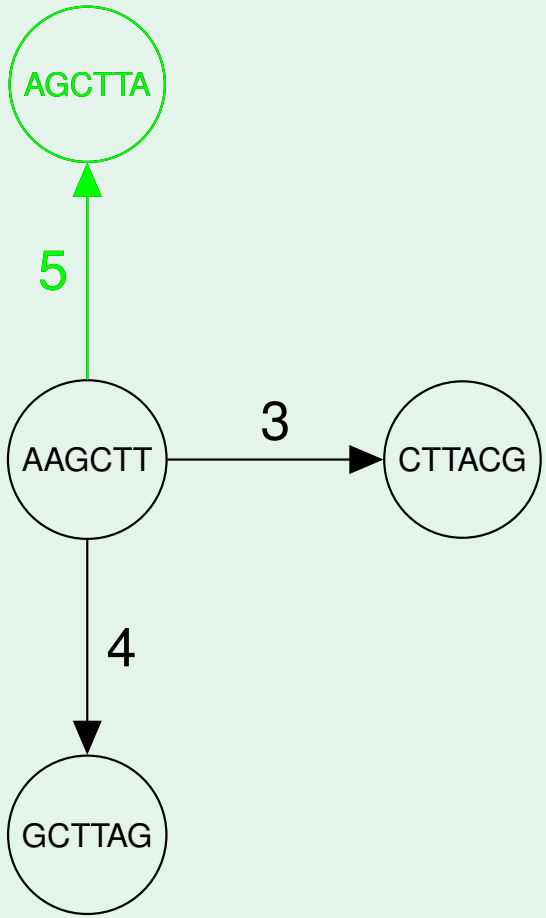
Example

k-mers set

1: **AA**GCTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index

{(1,2) ; (2,1) ; (5,0)}



Hybrid graph traversal

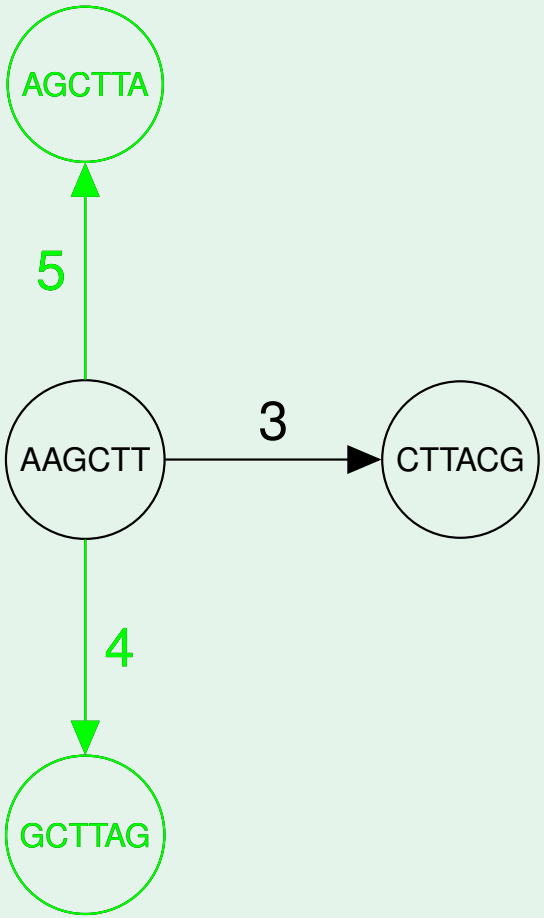
Example

k-mers set

1: **AA**GCTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

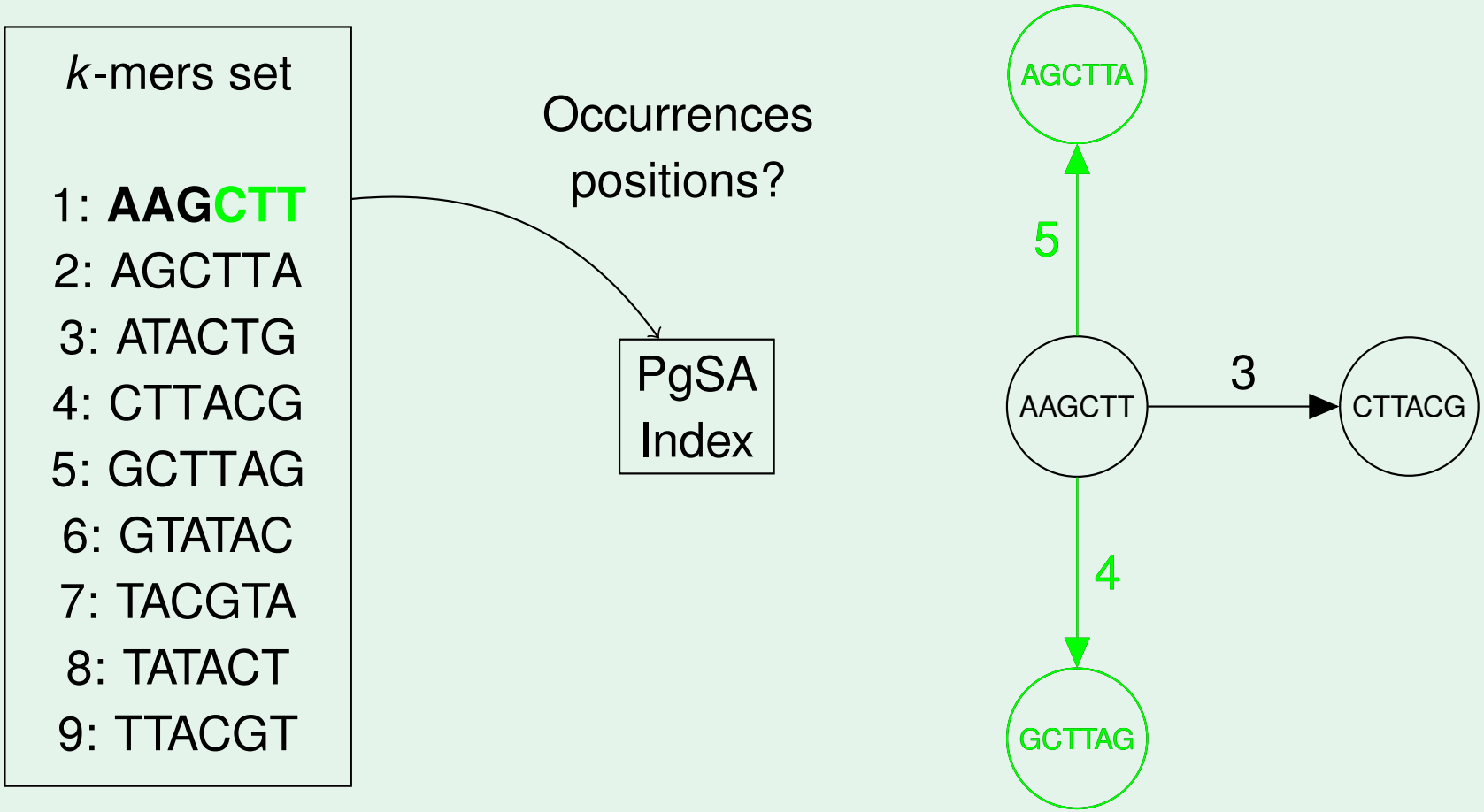
PgSA Index

{(1,2) ; (2,1) ; (5,0)}



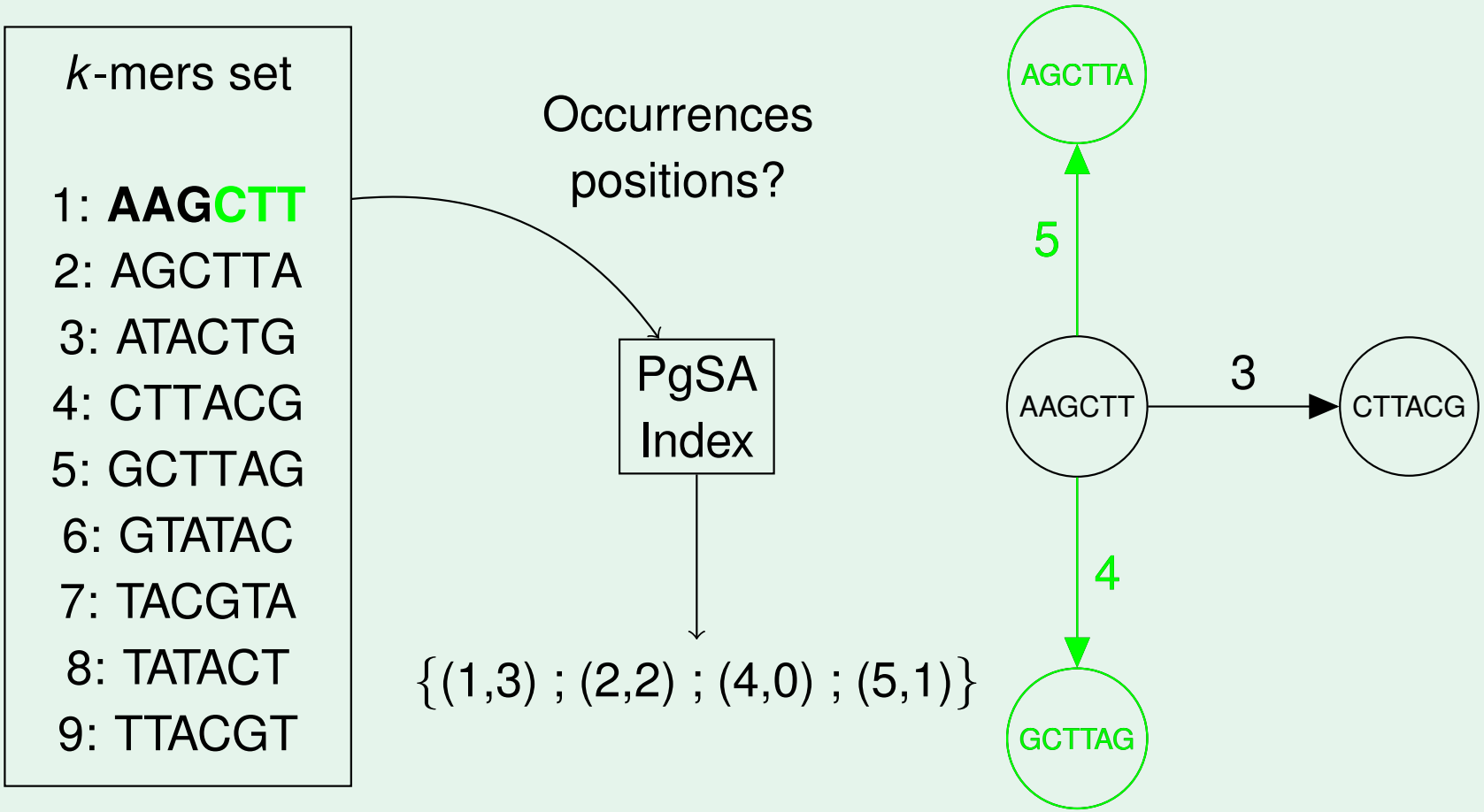
Hybrid graph traversal

Example



Hybrid graph traversal

Example



Hybrid graph traversal

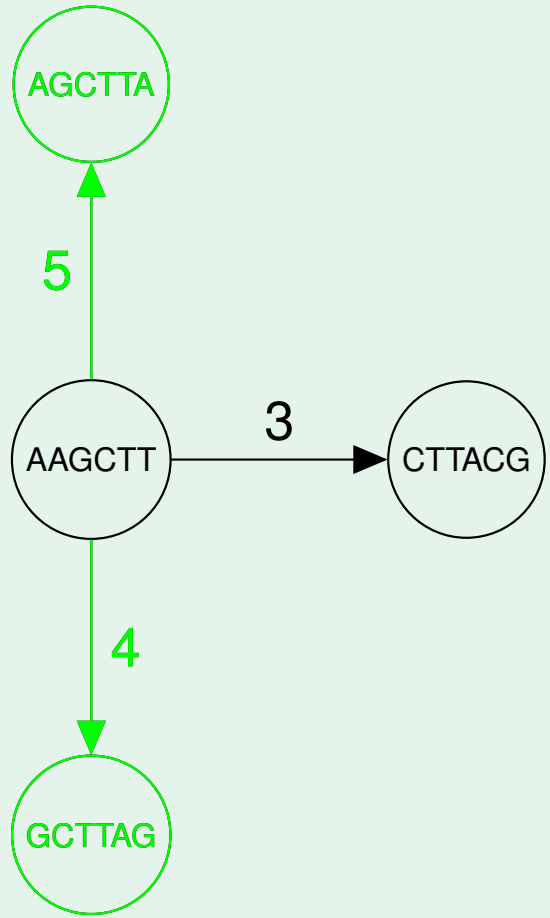
Example

k-mers set

1: **AAG**CTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index

{(1,3) ; (2,2) ; (4,0) ; (5,1)}



Hybrid graph traversal

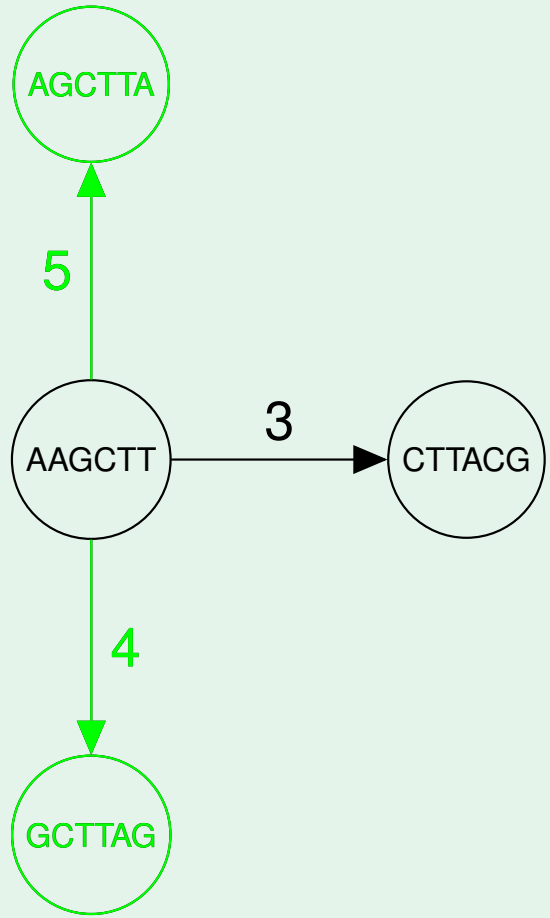
Example

k-mers set

1: **AAG**CTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index

{(1,3) ; (2,2) ; (4,0) ; (5,1)}



Hybrid graph traversal

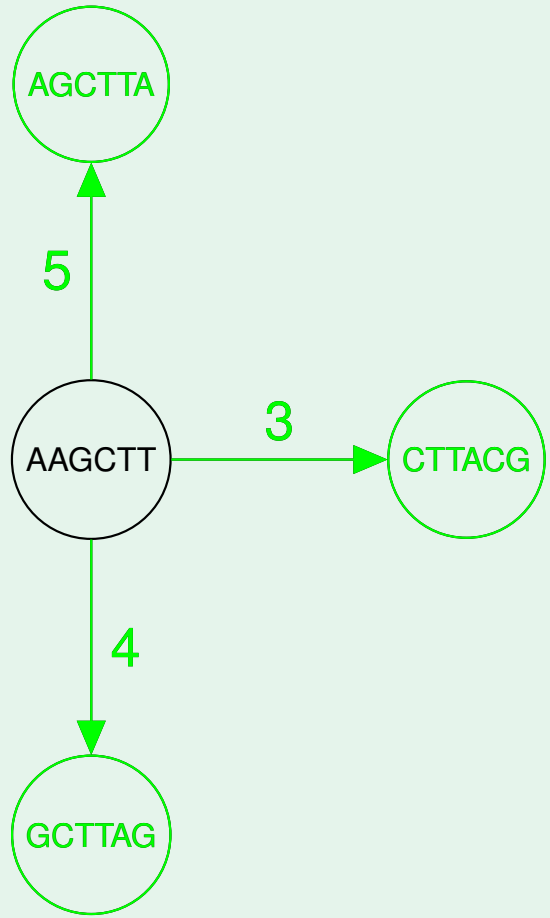
Example

k-mers set

1: **AAG**CTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index

{(1,3) ; (2,2) ; (4,0) ; (5,1)}



Hybrid graph traversal

Example

k-mers set

1: **AAG**CTT
2: AGCTTA
3: ATACTG
4: CTTACG
5: GCTTAG
6: GTATAC
7: TACGTA
8: TATACT
9: TTACGT

PgSA Index

{(1,3) ; (2,2) ; (4,0) ; (5,1)}