College of Computing and Information Technology, AASTMT
2017

DNA Sequencing Error Correction Algorithm

Student:

- *Salma Gomaa*

Supervisors:

- *Prof. Dr. Yasser El-sonbaty*
- *Dr. Nahla Belal*

Table of Contents

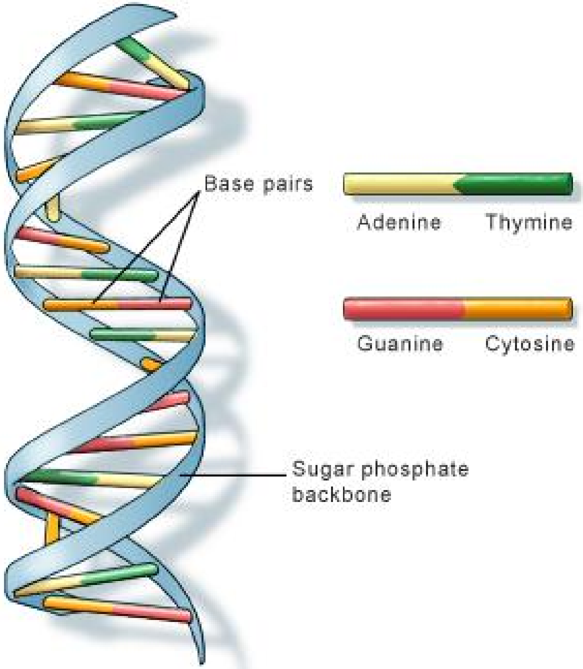
- Introduction
- Problem Definition
- Objective
- Related Work
- First Proposal and Defects
- H-RACER Proposal
- Evaluation
- Conclusion
- Future Work

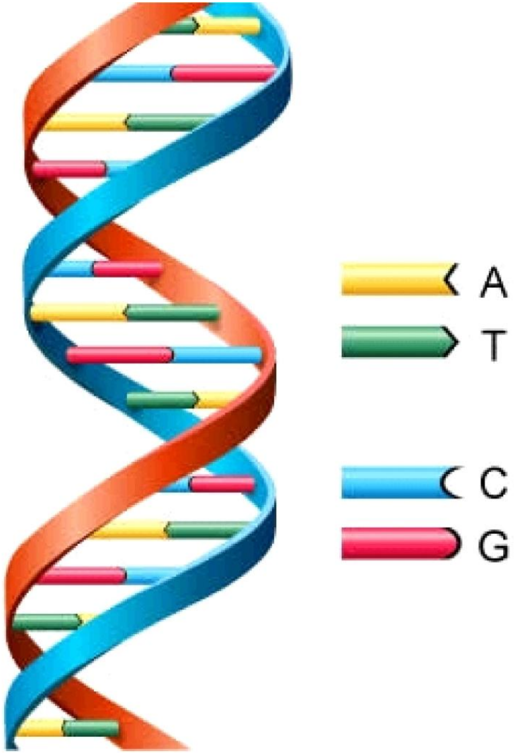
Table of Contents

- **Introduction**
- Problem Definition
- Objective
- Related Work
- First Proposal and Defects
- H-RACER Proposal
- Evaluation
- Conclusion
- Future Work

Deoxyribonucleic Acid (DNA)

Deoxyribonucleic Acid (DNA)





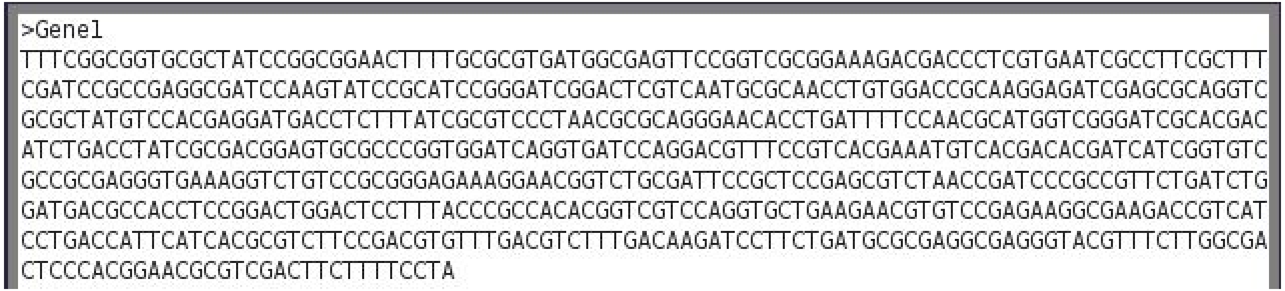
Deoxyribonucleic Acid (DNA)

➔ Human DNA consists of about 3 billion bases.

➔ The bases order determines the information available for building and maintaining an organism.

DNA Sequencing

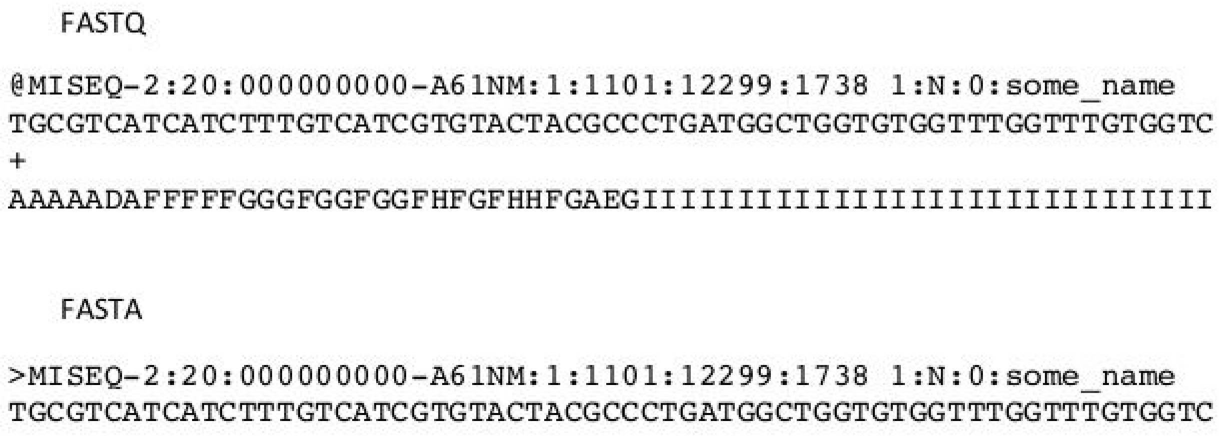
- ➔ DNA sequence - a single format onto which a broad range of biological phenomena can be projected for high-throughput data collection



```
>Gene1
TTTCGGCGGTGCGCTATCCGGCGGAACTTTTGCGCGTGATGGCGAGTTCCGGTCGCGGAAAGACGACCCTCGTGAATCGCCTTCGCTTT
CGATCCGCCGAGGCGATCCAAGTATCCGCATCCGGGATCGGACTCGTCAATGCGCAACCTGTGGACCGCAAGGAGATCGAGCGCAGGTC
GCGCTATGTCCACGAGGATGACCTCTTTATCGCGTCCCTAACGCGCAGGGAACACCTGATTTTCCAACGCATGGTCGGGATCGCACGAC
ATCTGACCTATCGCGACGGAGTGCGCCCGGTGGATCAGGTGATCCAGGACGTTTCCGTCACGAAATGTCACGACACGATCATCGGTGTC
GCCGCGAGGGTGAAAGGTCTGTCCGCGGGAGAAAGGAACGGTCTGCGATTCCGCTCCGAGCGTCTAACCGATCCCGCCGTTCTGATCTG
GATGACGCCACCTCCGGACTGGACTCCTTTACCCGCCACACGGTCGTCCAGGTGCTGAAGAACGTGTCCGAGAAGGCGAAGACCGTCAT
CCTGACCATTCATCACGCGTCTTCCGACGTGTTTGACGTCTTTGACAAGATCCTTCTGATGCGCGAGGCGAGGGTACGTTTCTTGGCGA
CTCCCACGGAACGCGTCGACTTCTTTTCCTA
```

DNA Sequencing

- DNA sequencing - outputs *fasta* or *fastq* file.



FASTQ

```
@MISEQ-2:20:000000000-A61NM:1:1101:12299:1738 1:N:0:some_name
TGCGTCATCATCTTTGTCATCGTGTACTACGCCCTGATGGCTGGTGTGGTTTGGTTTGTGGTC
+
AAAAADAFFFFFGGGFGGFGGFHFGFHHFGAEGIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
```

FASTA

```
>MISEQ-2:20:000000000-A61NM:1:1101:12299:1738 1:N:0:some_name
TGCGTCATCATCTTTGTCATCGTGTACTACGCCCTGATGGCTGGTGTGGTTTGGTTTGTGGTC
```

DNA Sequencing

- Next-generation DNA sequencing has the potential to dramatically accelerate biological and biomedical research

Next-Generation sequencing (NGS):

➔ NGS generates too many reads in a suitable time.

➔ NGS introduced two painful issues:
  - Read Length Shortness
  - Reads Accuracy Decrement

NGS Errors Corrections:

➔ Reads accuracy is a vital factor in all reads processes

➔ Detecting and Correcting errors is an essential step, and can be either:
  - Standalone Program
  - Process Preceding Step

NGS Errors Corrections:

- ➔ Detecting and Correcting errors depends on:
    - Nucleotide Frequency
    - Nucleotide Quality Value

- ➔ Nucleotide Error Types:
    - Substitution
    - Insertion
    - Deletion

NGS Errors - Substitution

- *Nucleotide Erroneous Substitution*

T	C	T	C	G

NGS Errors - Substitution

- *Nucleotide Erroneous Substitution*

T	C	T	C	G
T	C	A	C	G

NGS Errors - Substitution

➔ *Nucleotide Erroneous Substitution*

T	C	A	C	G

NGS Errors - Insertion

- ➔ *Erroneous Nucleotide Insertion*

T	C	T	C	G

NGS Errors - Insertion

➔ *Erroneous Nucleotide Insertion*

<table>
<tr><td>T</td><td>C</td><td colspan="2">T</td><td>C</td><td>G</td></tr>
<tr><td>T</td><td>C</td><td>G</td><td>T</td><td>C</td><td>G</td></tr>
</table>

NGS Errors - Insertion

➔ *Erroneous Nucleotide Insertion*

T	C	G	T	C	G

NGS Errors - Deletion

- *Nucleotide Deletion*

T	C	T	C	G

NGS Errors - Deletion

- *Nucleotide Deletion*

T	C	**T**	C	G
T	C	-	C	G

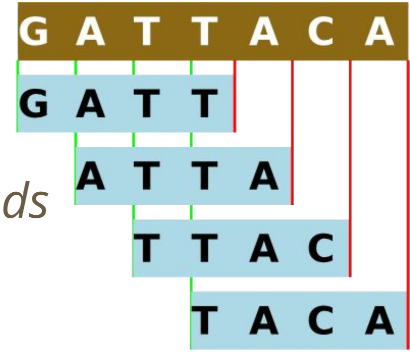
NGS Errors - Deletion

- *Nucleotide Deletion*

T	C	C	G

Correction Concepts and Definitions

- ➔ K-mer - *All the possible sub-sequences (of length k) from a read*



- ➔ K-mer Frequency - *Number of a k-mer repetition in all the reads*
- ➔ K-mer Frequency Threshold - *A preset threshold used in classifying k-mers*

Correction Concepts and Definitions

- Coverage - *Number of reads that include a given nucleotide in the sequence*

  C = LN/G

- Spectrum Alignment - *A filtration step that classifies the k-mers into strong and weak k-mers*

- Spectrum alignment depends on the k-mers frequencies and/or the nucleotides quality values

Correction Concepts and Definitions

➔ Correction can take place with:
- Spectrum Alignment - *by obtaining the nucleotides substitutions that leads to reduce the weak k-mers count*
- Tree Breadth-First Search - *by traversing multi out-going edges nodes, and removing fewer reads paths, then re-aligns them to the existing path*

Correction Concepts and Definitions

- Correction can take place with:
  - Reads Alignments - *by aligning reads with a common k-mer, then fixing misaligned nucleotides based on their occurrences and quality values*
  - Suffix Array - *built using a string of reads, and the correction takes place with the letter that appears most at each position*

Correction Concepts and Definitions

- Correction can take place with:
    - Suffix Trie - *the edges are labelled with DNA letters, where the correction is based on the number of leaves in the sub-trie rooted at the node*
    - K-mer Hashing Table - *by storing the total times each nucleotide appears before and after a k-mer, where the error is corrected via the counts*

Correction Concepts and Definitions

- ➔ Correction can take place with:
  - K-mer Discontinuities - *the frequencies of adjacent k-mers, where the correction is based on the removal or minimizing the discontinuity*

Evaluation Definitions

- ➔ True Positive - *Properly detected as erroneous and properly corrected*
- ➔ False Positive - *Improperly counted as erroneous and improperly corrected*
- ➔ False Negative - *Improperly considering as not erroneous*

Evaluation Definitions

- ➔ True Negative - *Properly considering as not erroneous*

- ➔ Sensitivity - *Ability to detect the erroneous nucleotides*

  Sensitivity = TP/(TP+FN)

Evaluation Definitions

➔ Specificity - *Ability to properly corrects the erroneous nucleotides*

Specificity = TN/(TN+FP)

➔ Accuracy - *All over error rate*

Accuracy = (TP+TN)/(TP+FP+FN+TN)

Table of Contents

- Introduction
- **Problem Definition**
- Objective
- Related Work
- First Proposal and Defects
- H-RACER Proposal
- Evaluation
- Conclusion
- Future Work

Problem Definition

- ➔ DNA Next Sequencing Generation - *generates reads with many errors with different types*

- ➔ DNA Reads Accuracy - *is a vital factor in all of the DNA reads processes*

Table of Contents

- Introduction
- Problem Definition
- **Objective**
- Related Work
- First Proposal and Defects
- H-RACER Proposal
- Evaluation
- Conclusion
- Future Work

Objective

- ➔ Raising the DNA reads accuracy
- ➔ Correcting all of the different types of errors
- ➔ Accomplishing the correction process within the shortest time

Table of Contents

- Introduction
- Problem Definition
- Objective
- **Related Work**
- First Proposal and Defects
- H-RACER Proposal
- Evaluation
- Conclusion
- Future Work

Related Work

- ➔ Correction Methodologies can be classified into two categories:
  - Substitution only correction
  - Substitution, insertion and deletions correction

Related Work

- ➔ Substitution only correction methodology, like:

Euler	Velvet	AllPaths	SOAP	Quake	Reptile	CUDA	HiTEC	RACER	EC
2004	2008	2008	2010	2010	2010	2010	2011	2013	2015
Spectrum Alignment	Tree BFS	Spectrum Alignment	Tree BFS	Spectrum Alignment	Spectrum Alignment	Spectrum Alignment	Suffix Array	Hash Table	Hash Table
K-mer Freq.	Nuc. Freq.	K-mer Freq. with Nuc. QV	K-mer Freq.	K-mer Freq. with Nuc. QV	K-mer Freq. with Nuc. QV	K-mer Freq. with Votes	Nuc. Freq.	Nuc. Freq.	Nuc. Freq.

Related Work

➔ Substitution, insertion and deletions correction methodology , like:

HSHREC	Coral	Pollx
2010	2011	2015
Suffix Trie	Reads Alignment	K-mer Discontinuities
Nuc. Freq.	Nuc. Freq. with QV	K-mer Freq.

Related Work

- RACER characteristics:
  - Ability to correct data sets that have varying read lengths
  - Hash table and k-mer nucleotides neighbours
  - Fastest DNA error correction algorithm existent nowadays with a high accuracy
  - Corrects substitutions only

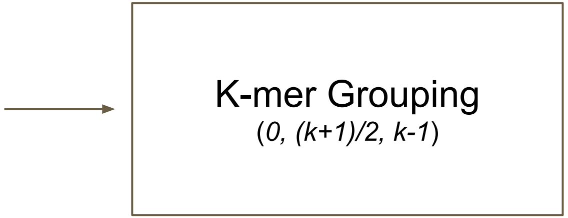
Table of Contents

- Introduction
- Problem Definition
- Objective
- Related Work
- **First Proposal and Defects**
- H-RACER Proposal
- Evaluation
- Conclusion
- Future Work

First Proposal and Defects

- ➔ Aiming to correct all types of errors
- ➔ Hashing the k-mers into integers
- ➔ Flexible to run more correction iterations

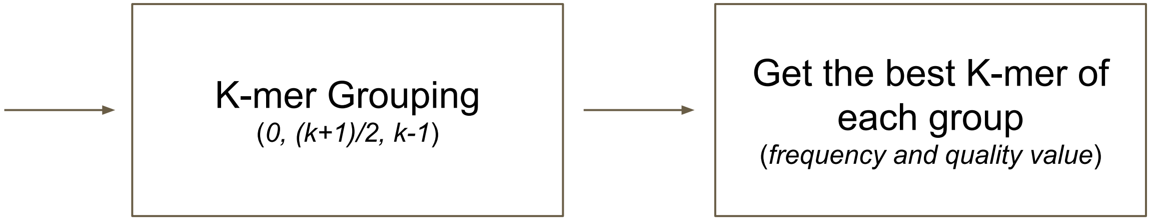
First Proposal and Defects



K-mer Grouping

(0, (k+1)/2, k-1)

First Proposal and Defects



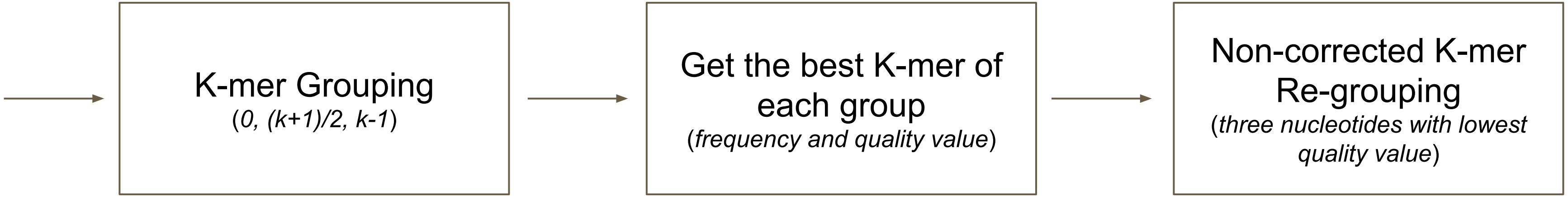
K-mer Grouping

(*0, (k+1)/2, k-1*)

Get the best K-mer of each group

(*frequency and quality value*)

First Proposal and Defects



K-mer Grouping

(0, (k+1)/2, k-1)

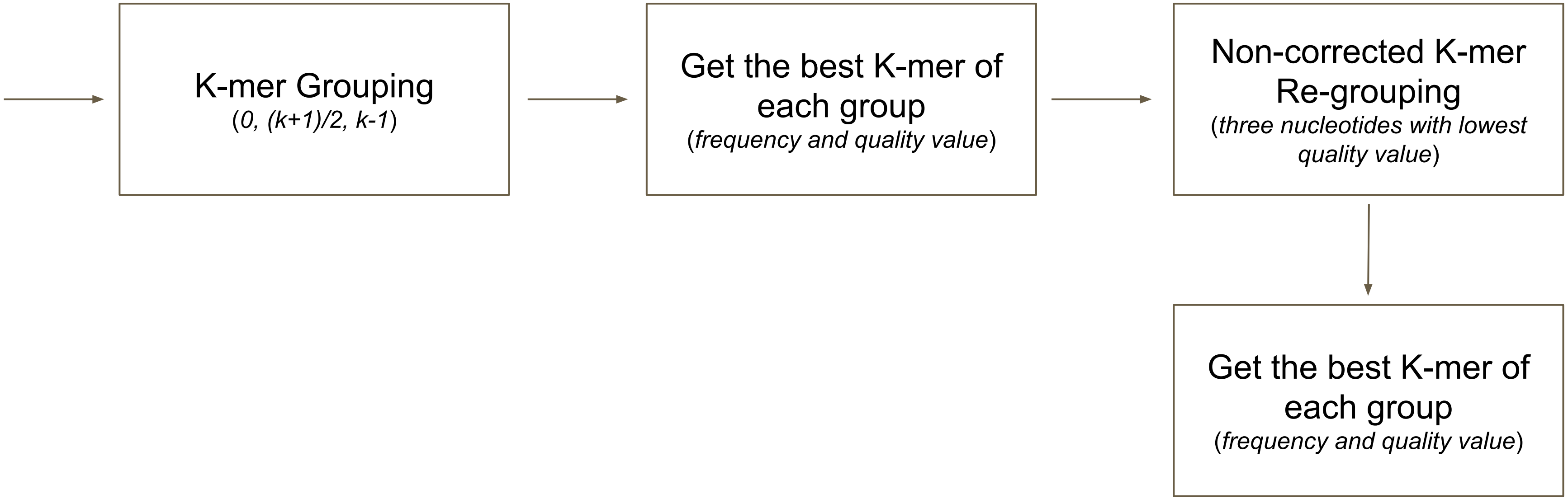
Get the best K-mer of each group

(*frequency and quality value*)

Non-corrected K-mer Re-grouping

(*three nucleotides with lowest quality value*)

First Proposal and Defects



K-mer Grouping
(0, (k+1)/2, k-1)

Get the best K-mer of each group
(*frequency and quality value*)

Non-corrected K-mer Re-grouping
(*three nucleotides with lowest quality value*)

Get the best K-mer of each group
(*frequency and quality value*)

First Proposal and Defects

➔ Data sets

Name	Genome Length	Read Length	Number of Reads	Coverage
Lactococcus Lactis	2,598,144	36	4,370,050	60.55

First Proposal - Evaluation

➔ Lactococcus Lactis (G: 2,598,144 - L: 36, N: 4,370,050, C: 60.55)

	Coral	Pollux	HSHSREC	First Proposal
Accuracy in Percentage	91.45	94.15	95.34	95.39
Time in Minutes	5	3	15	61

First Proposal and Defects

- ➔ This algorithm is mainly dependent on the k-mers grouping

- ➔ kmers grouping takes place by generating all of the possible cases of the corrections of every kmer, and here goes the time defect (exponential)

- ➔ On removing the method with the exponential complexity, the accuracy of the algorithm has been greatly negatively affected.

First Proposal and Defects

➔ The main major step of the proposal implies to it's weakness point, which proves that this proposal won't get a better results

➔ So, it fails to run on big data

➔ Using real data sets to get a good indication of real life performance

Table of Contents

- Introduction
- Problem Definition
- Objective
- Related Work
- First Proposal and Defects
- **H-RACER Proposal**
- Evaluation
- Conclusion
- Future Work

H-RACER Proposal

- H-RACER is a newly correction approach for correcting all types of errors.

- H-RACER is inherited from RACER

- RACER is the fastest algorithm specialized in correction substitution errors only

H-RACER Proposal

➔ RACER characteristics:

- Ability to correct data sets that have varying read lengths
- Hash table and k-mer nucleotides neighbours
- Fastest DNA error correction algorithm existent nowadays with a high accuracy
- Corrects substitutions only

H-RACER Proposal

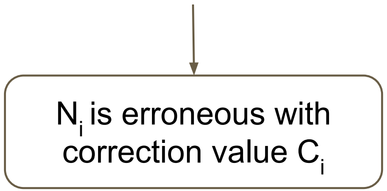
- ➔ H-RACER uses the same algorithm of RACER in detecting errors and deciding corrections values

- ➔ H-RACER detects the error type for an erroneous nucleotide by studying its correction value against its neighbours

- ➔ H-RACER decides the corrective action (substitute, insert, delete) according to the detected error type

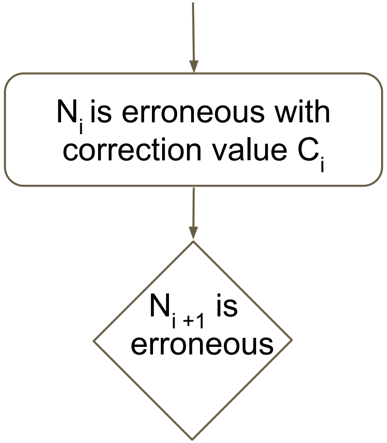
H-RACER - Error Type Detection

H-RACER - Error Type Detection



N_i is erroneous with correction value C_i

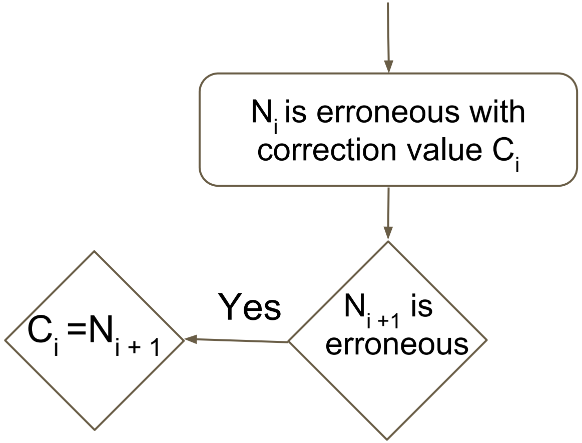
H-RACER - Error Type Detection



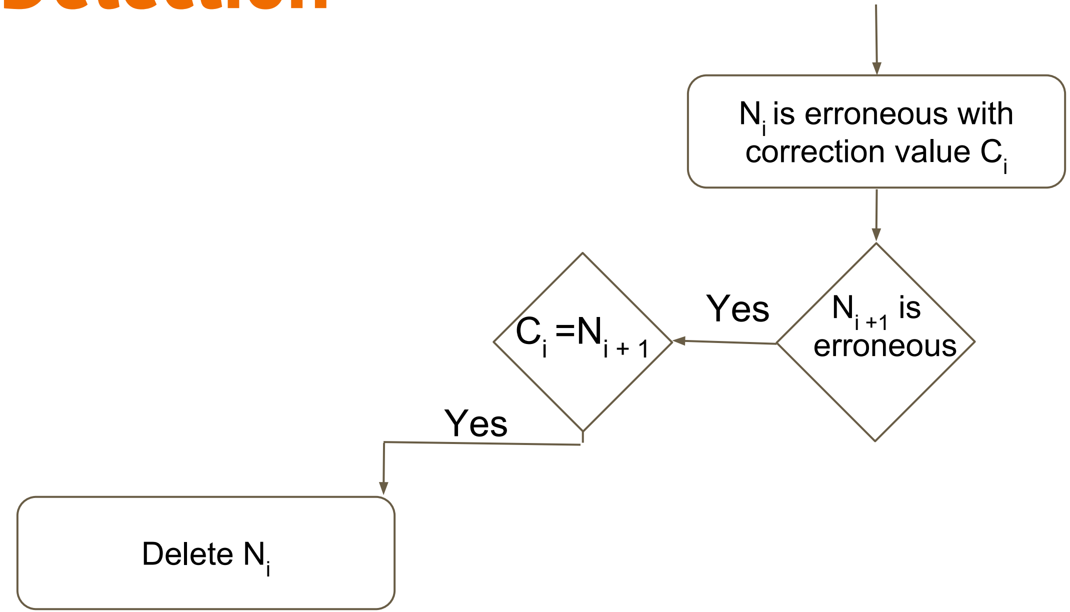
N_i is erroneous with correction value C_i

N_{i+1} is erroneous

H-RACER - Error Type Detection



H-RACER - Error Type Detection



N_i is erroneous with correction value C_i

N_{i+1} is erroneous

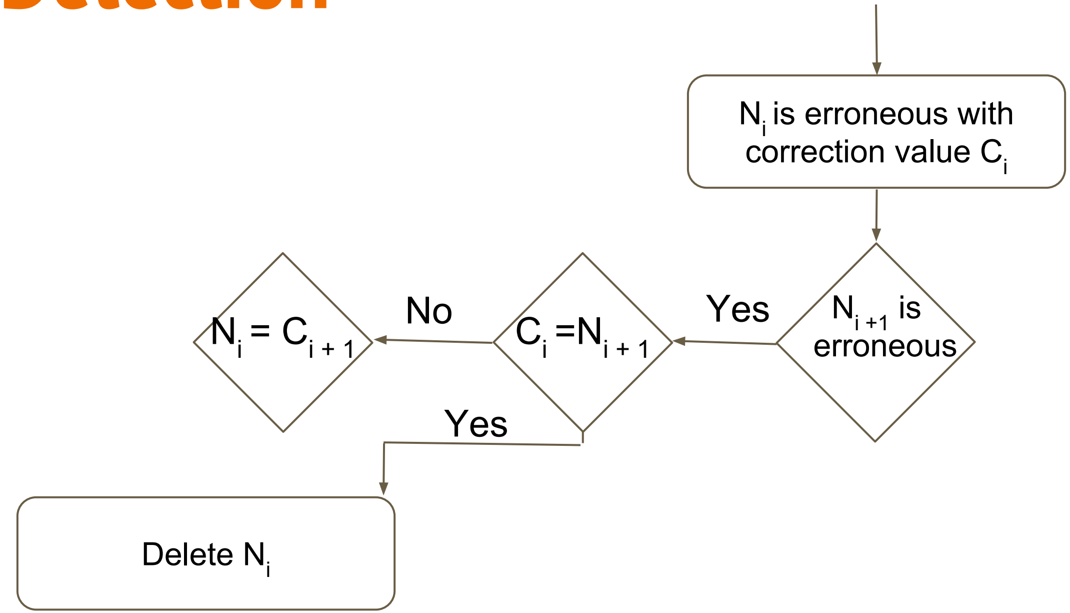
Yes

$C_i = N_{i+1}$

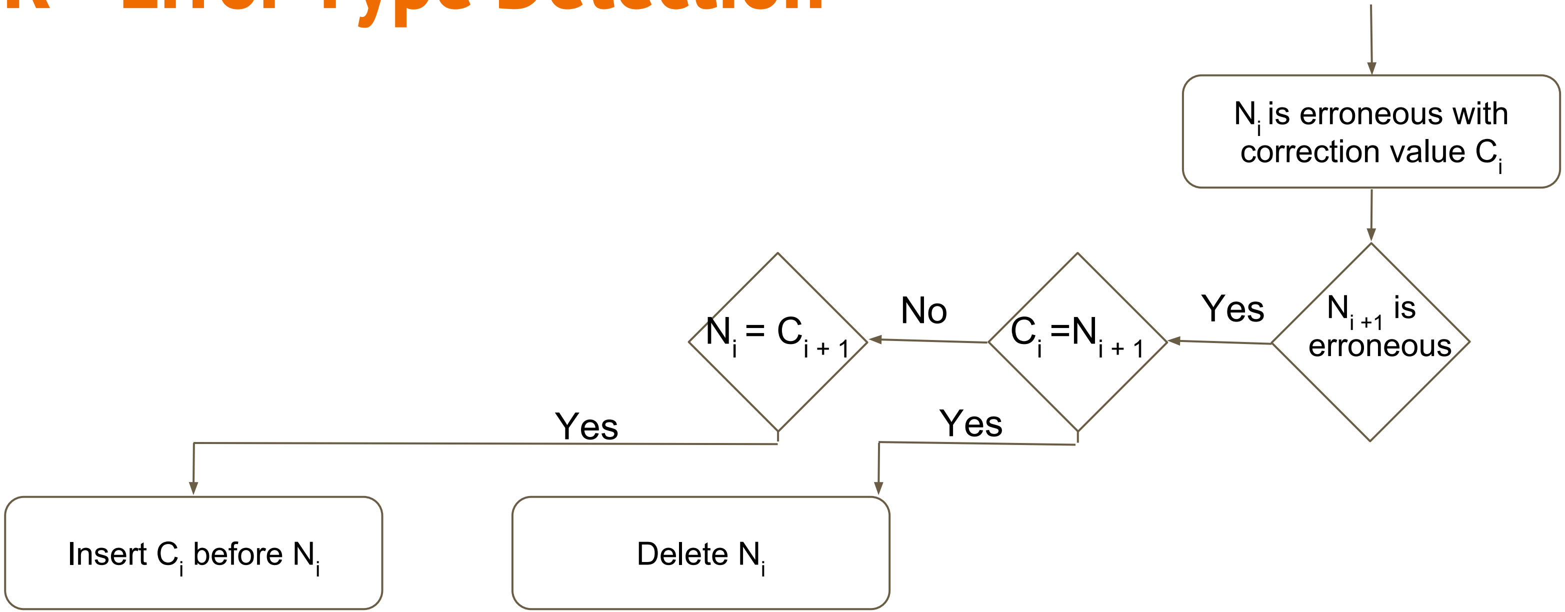
Yes

Delete N_i

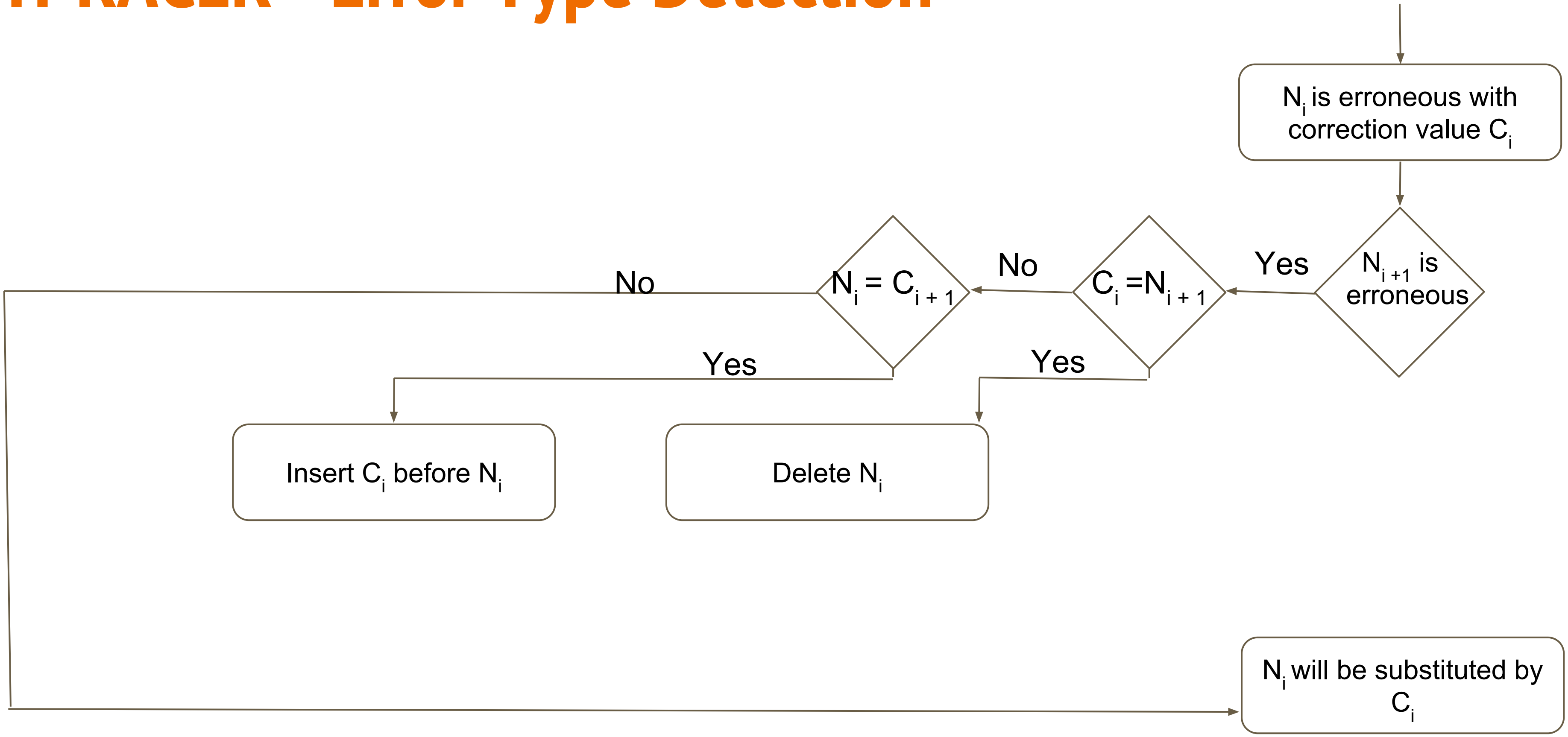
H-RACER - Error Type Detection



H-RACER - Error Type Detection

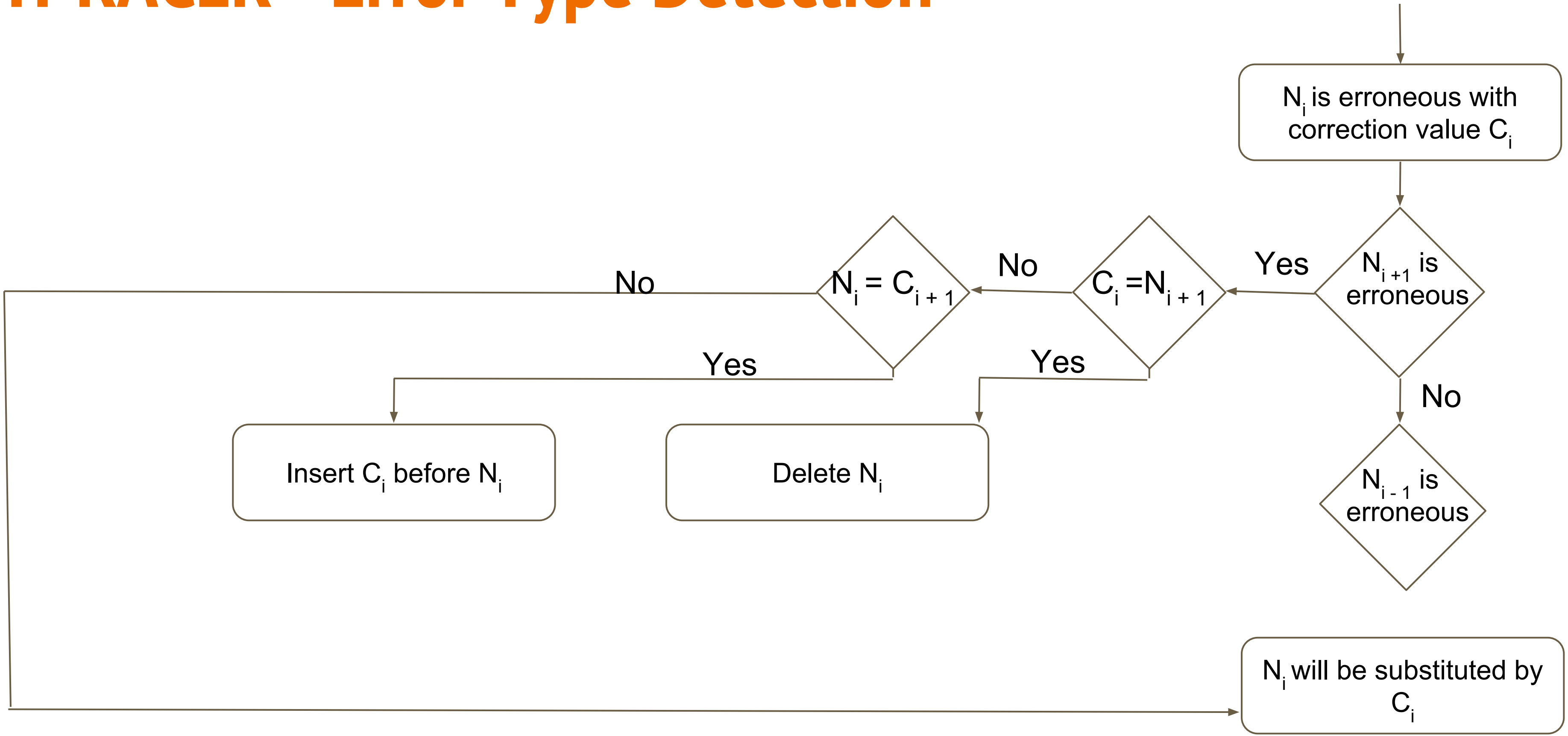


H-RACER - Error Type Detection



N_i is erroneous with correction value C_i
N_{i+1} is erroneous
Yes
$C_i = N_{i+1}$
No
$N_i = C_{i+1}$
No
Yes
Yes
Insert C_i before N_i
Delete N_i
N_i will be substituted by C_i

H-RACER - Error Type Detection



N_i is erroneous with correction value C_i

N_{i+1} is erroneous

Yes

$C_i = N_{i+1}$

No

$N_i = C_{i+1}$

No

Yes

Yes

No

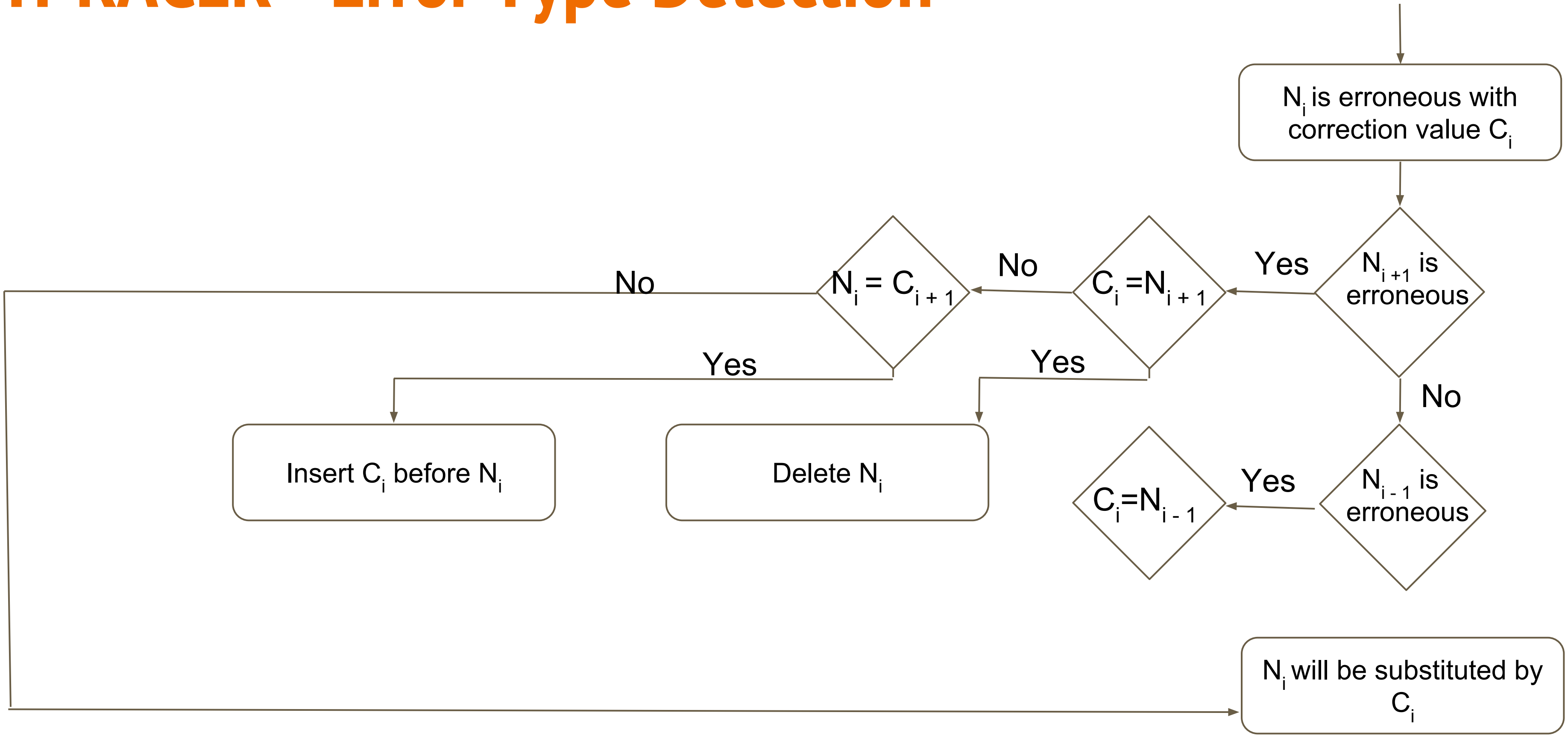
N_{i-1} is erroneous

Insert C_i before N_i

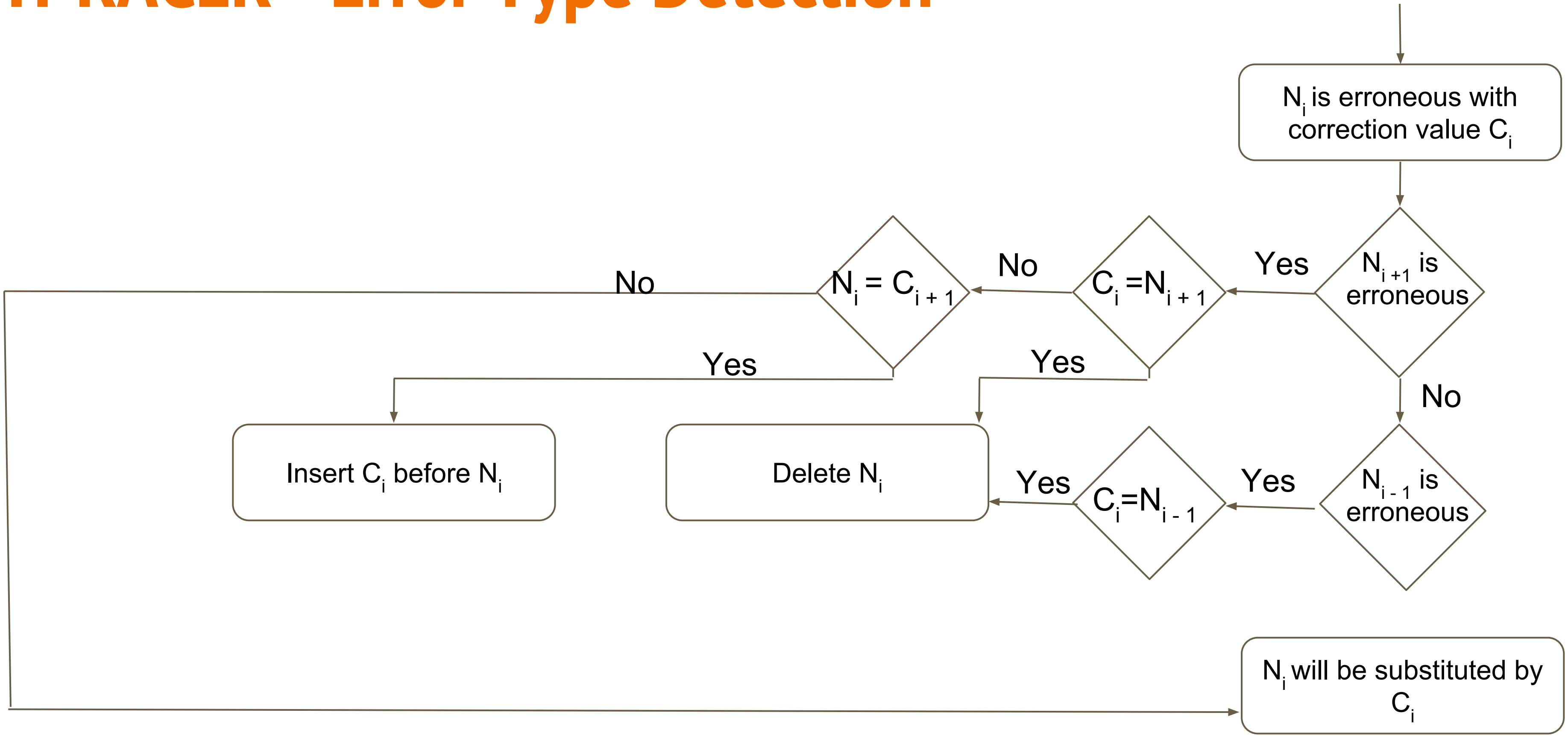
Delete N_i

N_i will be substituted by C_i

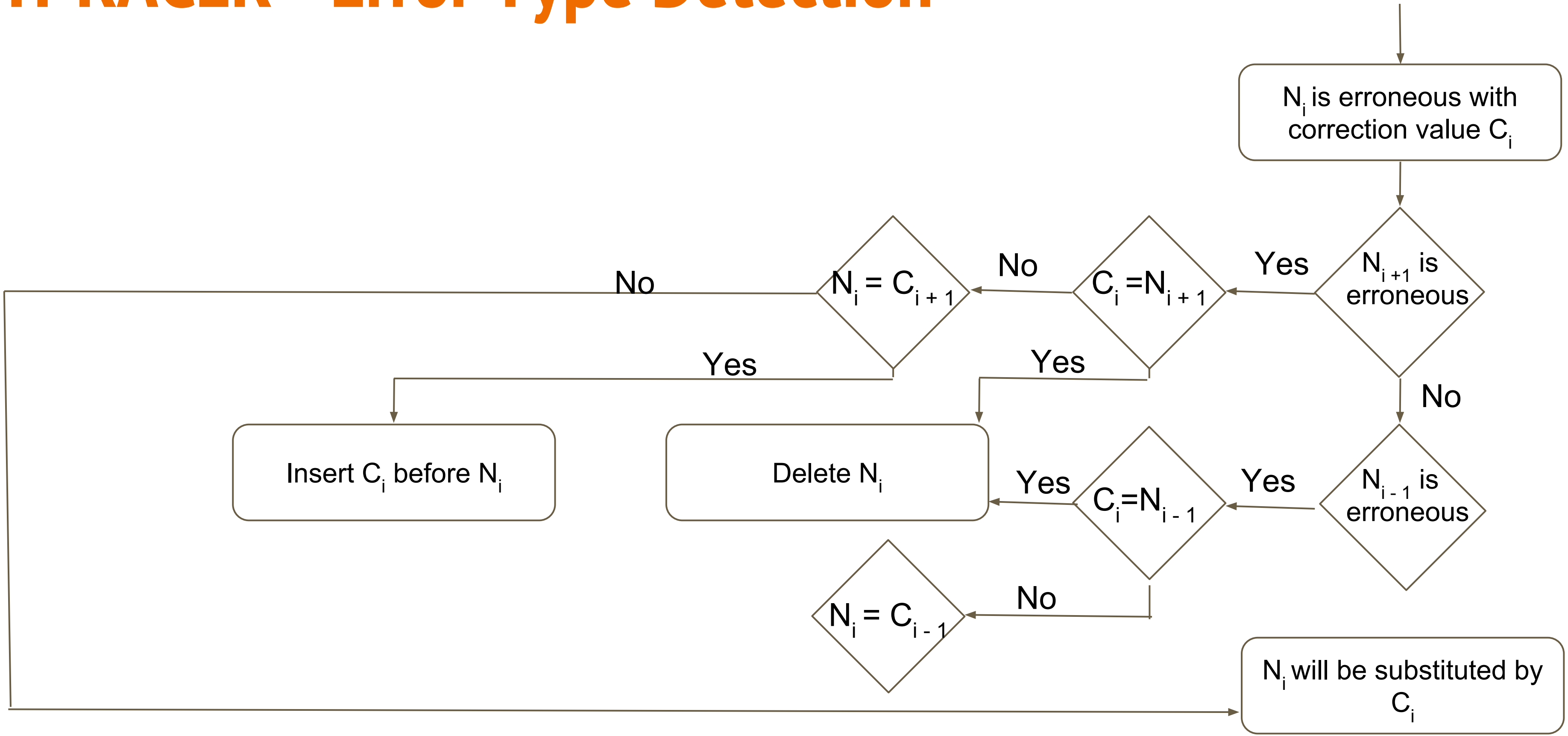
H-RACER - Error Type Detection



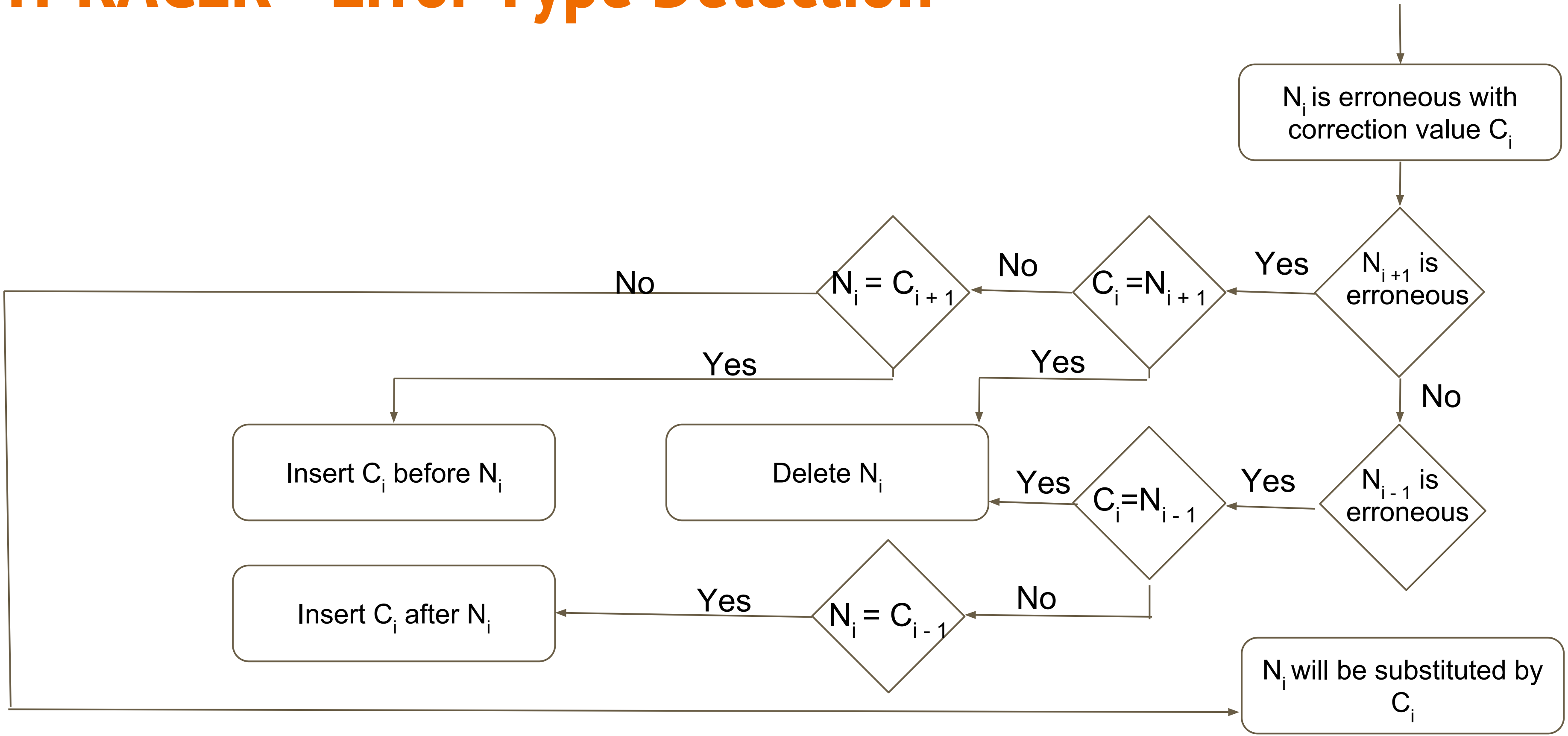
H-RACER - Error Type Detection



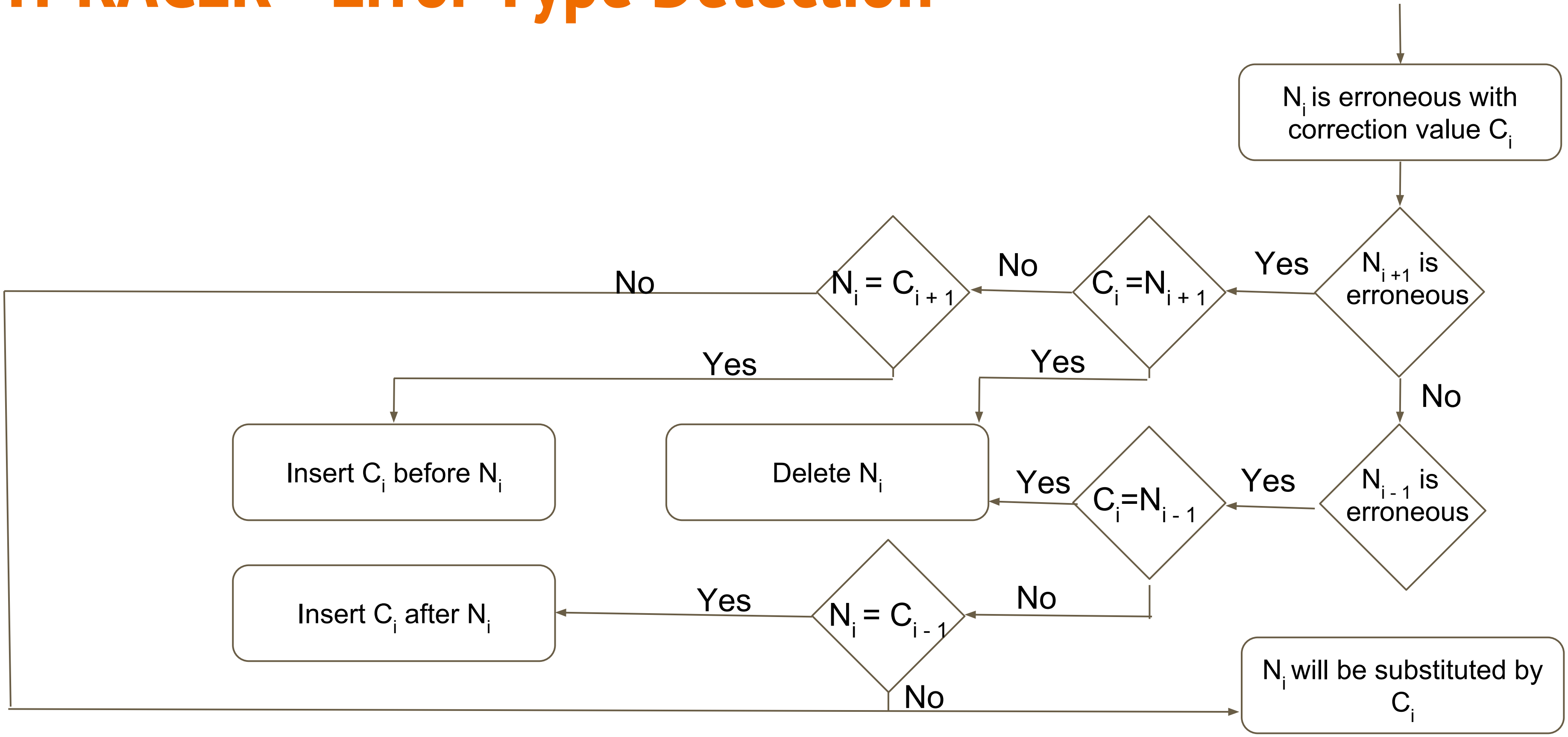
H-RACER - Error Type Detection



H-RACER - Error Type Detection



H-RACER - Error Type Detection



N_i is erroneous with correction value C_i

N_{i+1} is erroneous

$C_i = N_{i+1}$

$N_i = C_{i+1}$

N_{i-1} is erroneous

$C_i = N_{i-1}$

$N_i = C_{i-1}$

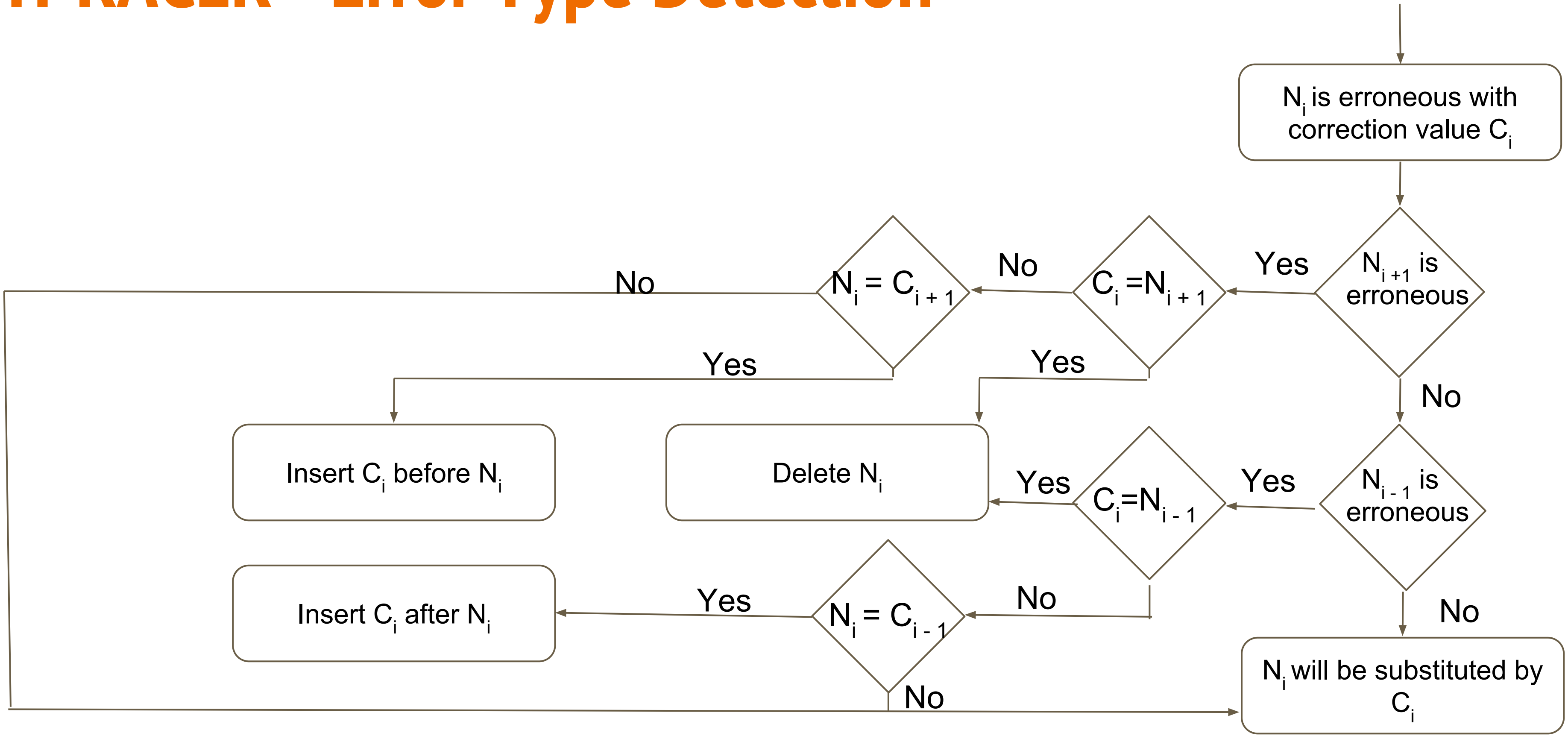
Insert C_i before N_i

Delete N_i

Insert C_i after N_i

N_i will be substituted by C_i

H-RACER - Error Type Detection



N_i is erroneous with correction value C_i

N_{i+1} is erroneous

Yes

$C_i = N_{i+1}$

No

$N_i = C_{i+1}$

No

Yes

Yes

Insert C_i before N_i

Delete N_i

No

N_{i-1} is erroneous

Yes

$C_i = N_{i-1}$

Yes

No

$N_i = C_{i-1}$

Yes

Insert C_i after N_i

No

No

N_i will be substituted by C_i

H-RACER - Error Type Detection

➔ Erroneously Inserted Nucleotide

...ACCATG...

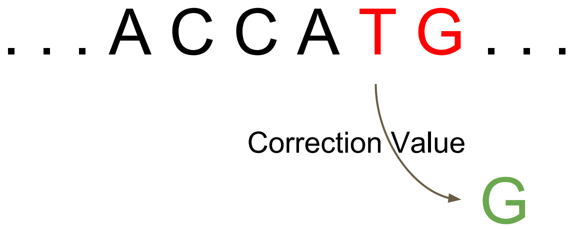
H-RACER - Error Type Detection

➔ Erroneously Inserted Nucleotide

. . . A C C A T G . . .

H-RACER - Error Type Detection

➔ Erroneously Inserted Nucleotide



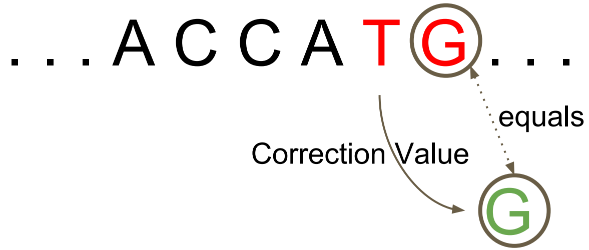
. . . A C C A T G . . .

Correction Value

G

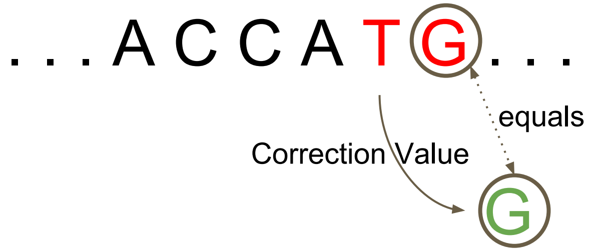
H-RACER - Error Type Detection

➔ Erroneously Inserted Nucleotide



H-RACER - Error Type Detection

➔ Erroneously Inserted Nucleotide



. . . A C C A T G . . .

equals

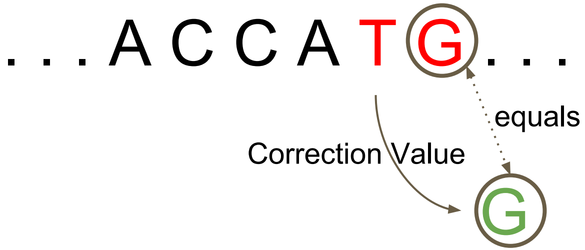
Correction Value

G

∴ T - *erroneously inserted nucleotide*

H-RACER - Error Type Detection

➔ Erroneously Inserted Nucleotide



. . . A C C A T G . . .

equals

Correction Value

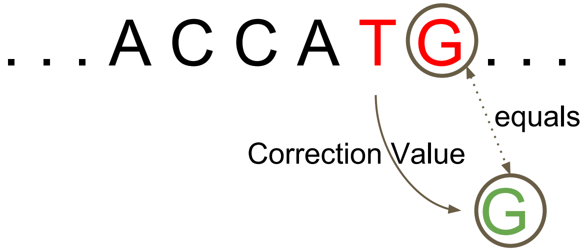
G

∴ T - *erroneously inserted nucleotide*

∴ Correction - *delete* T

H-RACER - Error Type Detection

- Erroneously Inserted Nucleotide



...ACCATG...

Correction Value

equals

G

∴ T - *erroneously inserted nucleotide*

∴ Correction - *delete* T

...ACCA**G**...

H-RACER - Error Type Detection

- ➔ Erroneously Deleted Nucleotide

. . . A A C C T G . . .

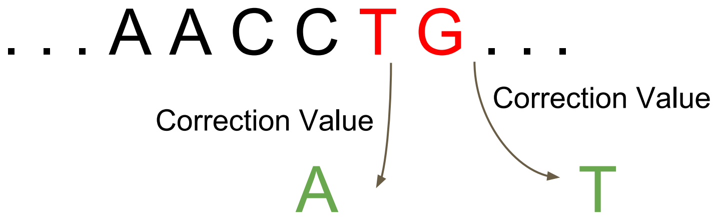
H-RACER - Error Type Detection

➔ Erroneously Deleted Nucleotide

. . . A A C C T G . . .

H-RACER - Error Type Detection

➔ Erroneously Deleted Nucleotide



. . . A A C C T G . . .

Correction Value

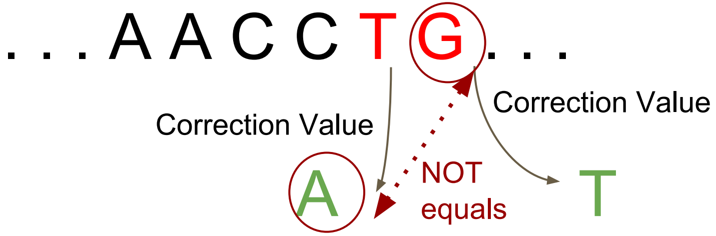
A

Correction Value

T

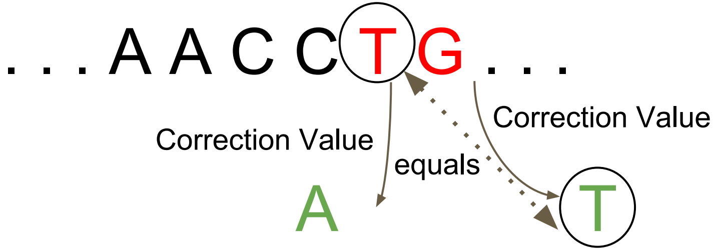
H-RACER - Error Type Detection

➔ Erroneously Deleted Nucleotide



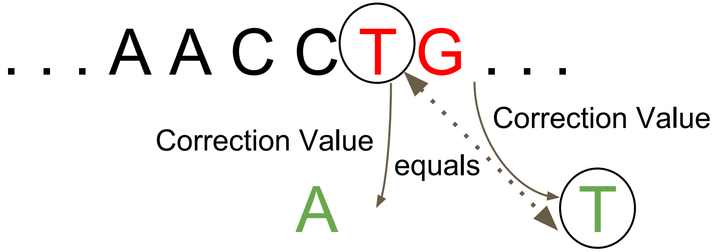
H-RACER - Error Type Detection

➔ Erroneously Deleted Nucleotide



H-RACER - Error Type Detection

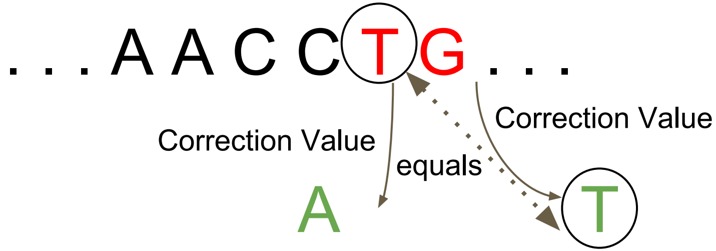
- ➔ Erroneously Deleted Nucleotide



∴ A - *erroneously deleted nucleotide*

H-RACER - Error Type Detection

- ➔ Erroneously Deleted Nucleotide



. . . A A C C T G . . .

Correction Value

equals

Correction Value

A

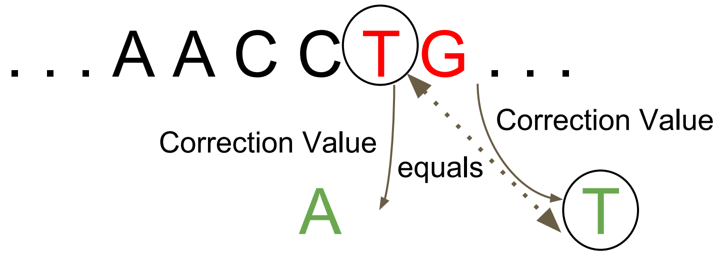
T

∴ A - *erroneously deleted nucleotide*

∴ Correction - *insert* A

H-RACER - Error Type Detection

➔ Erroneously Deleted Nucleotide



. . . A A C C T G . . .

Correction Value

equals

Correction Value

A

T

∴ A - *erroneously deleted nucleotide*

∴ Correction - *insert* A

. . . A A C C **A** T G . . .

H-RACER - Error Type Detection

- ➔ Erroneously Substituted Nucleotide

. . . A A C C T G . . .

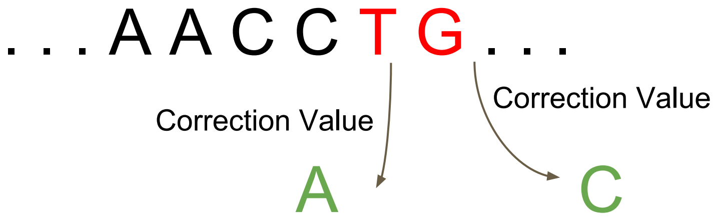
H-RACER - Error Type Detection

- ➔ Erroneously Substituted Nucleotide

. . . A A C C T G . . .

H-RACER - Error Type Detection

➔ Erroneously Substituted Nucleotide



. . . A A C C T G . . .

Correction Value

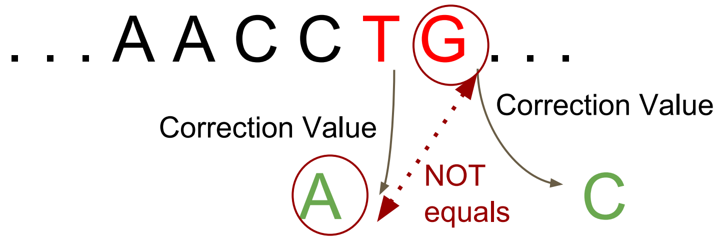
A

Correction Value

C

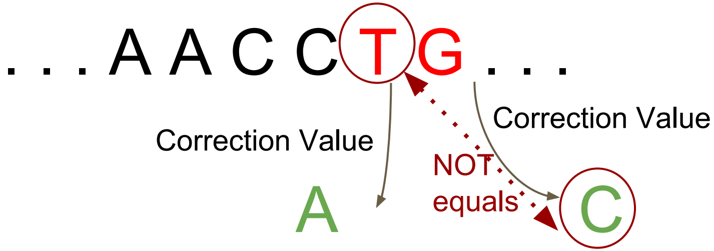
H-RACER - Error Type Detection

➔ Erroneously Substituted Nucleotide



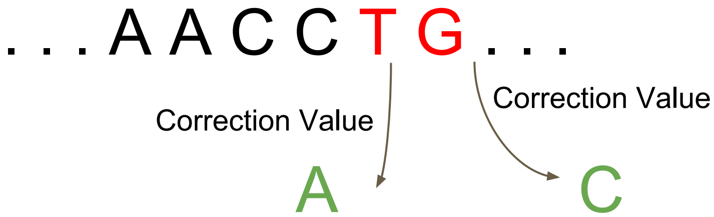
H-RACER - Error Type Detection

- ➔ Erroneously Substituted Nucleotide



H-RACER - Error Type Detection

➔ Erroneously Substituted Nucleotide



. . . A A C C T G . . .

Correction Value

A

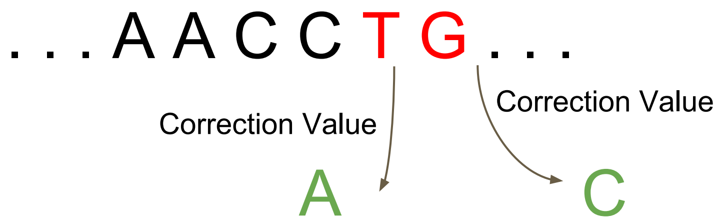
Correction Value

C

∴ A - *erroneously substituted by* T, *and,* C - *erroneously substituted by* G

H-RACER - Error Type Detection

➔ Erroneously Substituted Nucleotide

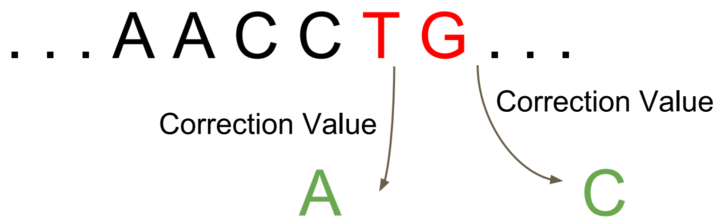


∴ A - *erroneously substituted by* T*, and,* C - *erroneously substituted by* G

∴ Correction - *substitute* T by A *and* G by C

H-RACER - Error Type Detection

➔ Erroneously Substituted Nucleotide



. . . A A C C T G . . .

Correction Value

Correction Value

A

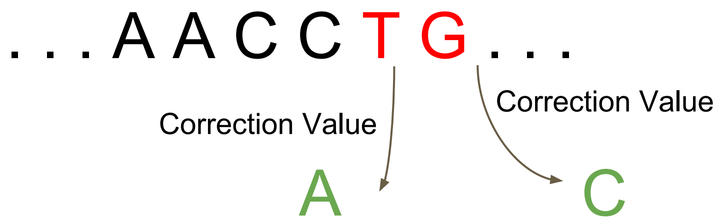
C

∴ A - *erroneously substituted by* T, *and,* C - *erroneously substituted by* G

∴ Correction - *substitute* T by A *and* G by C

H-RACER - Error Type Detection

➔ Erroneously Substituted Nucleotide



. . . A A C C T G . . .

Correction Value

Correction Value

A

C

∴ A - *erroneously substituted by* T, *and,* C - *erroneously substituted by* G

∴ Correction - *substitute* T by A *and* G by C

. . . A A C C **A C** . . .

Table of Contents

- Introduction
- Problem Definition
- Objective
- Related Work
- First Proposal and Defects
- H-RACER Proposal
- **Evaluation**
- Conclusion
- Future Work

H-RACER - Evaluation

- ➔ Data sets were brought from the National Center for Biotechnology Information (NCBI)

- ➔ Executing on amazon elastic cloud (AWS EC2) instance with 32 vCPU and 244GiB RAM, with Linux (Ubuntu) operating system

- ➔ Verified by a standalone C/C++ program implemented by RACER, that has the advantage of avoiding the interference of mapping/assembling programs

H-RACER - Evaluation

- Data sets

Name	Genome Length	Read Length	Number of Reads	Coverage
Lactococcus Lactis	2,598,144	36	4,370,050	60.55
Treponema Pallidum	1,139,417	35	7,133,663	219.13
E.coli 75a	4,639,675	75	3,454,048	55.83
E.coli 75b	4,639,675	75	4,341,061	70.17

H-RACER - Evaluation

➔ Lactococcus Lactis

	Coral	Pollux	HSHSREC	H-RACER
True Positive in Millions	15.4	25.3	25.5	21.2
False Positive in Millions	2.0	7.7	6.1	0.02
False Negative in Millions	11.4	1.5	1.3	5.6
True Negative in Millions	128.5	122.8	124.5	130.5
Sensitivity in Percentage	57.43	94.46	95.25	79.22
Specificity in Percentage	98.44	94.08	95.36	99.98
Accuracy in Percentage	91.45	94.15	95.34	96.45
Time in Minutes	5	3	15	1

H-RACER - Evaluation

➔ Treponema Pallidum

	Coral	Pollux	HSHSREC	H-RACER
True Positive in Millions	25.60	63.9	64.4	56.3
False Positive in Millions	3.5	8.8	8.1	0.2
False Negative in Millions	41.6	3.3	2.7	10.8
True Negative in Millions	179.1	173.7	174.4	182.4
Sensitivity in Percentage	38.08	95.15	95.95	83.87
Specificity in Percentage	98.10	95.16	95.55	99.88
Accuracy in Percentage	81.97	95.16	95.65	95.58
Time in Minutes	12	3	22	2

H-RACER - Evaluation

➔ E.coli 75a

	Coral	Pollux	HSHSREC	H-RACER
True Positive in Millions	26.4	80.0	N/A	76.3
False Positive in Millions	5.6	31.7	N/A	0.03
False Negative in Millions	73.7	20.2	N/A	23.8
True Negative in Millions	153.4	127.2	N/A	158.9
Sensitivity in Percentage	26.40	79.87	N/A	76.21
Specificity in Percentage	96.51	80.07	N/A	99.98
Accuracy in Percentage	69.40	79.99	N/A	90.79
Time in Minutes	9	16	N/A	1

H-RACER - Evaluation

➔ E.coli 75b

	Coral	Pollux	HSHSREC	H-RACER
True Positive in Millions	13.3	99.4	N/A	81.06
False Positive in Millions	3.7	37.8	N/A	0.04
False Negative in Millions	108.5	22.4	N/A	40.8
True Negative in Millions	200.0	166.0	N/A	203.7
Sensitivity in Percentage	10.93	81.58	N/A	66.54
Specificity in Percentage	98.19	81.46	N/A	99.98
Accuracy in Percentage	65.55	81.50	N/A	87.47
Time in Minutes	13	21	N/A	2

H-RACER - Evaluation

➔ H-RACER has the best results in accuracy and time, especially for long genomes

➔ H-RACER uses the bitwise orientation in implementation (inherited from RACER), so it shows the best time

➔ H-RACER error detection algorithm has a complexity O(*r*), where *r* is the number of reads

H-RACER - Evaluation

- H-RACER has the best accuracy, as it depends on:
  - Lowering false positive rate
  - Lowering sensitivity rate

- Lowering false positive rate negatively affects the true positive and false negative rates

H-RACER - Evaluation

- ➔ Enhancing the reads overall accuracy is the main vital target. So, corrective algorithms should not introduce errors (represented in false positive rate).

- ➔ H-RACER has the best accuracy although it hasn’t the best sensitivity rate

H-RACER - Evaluation

➔ Using genomes with high coverage rate, negatively affects H-RACER by:

- Increasing error detection ambiguity
- Raising false negative rate
- Lowering accuracy

H-RACER - Evaluation

- ➔ The comparisons were established between H-RACER and algorithms specialized in correcting all types of errors

- ➔ Using real data sets, to get a good indication of real life performance

- ➔ Different data sets, with different read length, genome size and coverage

H-RACER - Evaluation

- Data sets were brought from the National Center for Biotechnology Information (NCBI)

- Executing on amazon elastic cloud (AWS EC2) instance with 32 vCPU and 244GiB RAM, with Linux (Ubuntu) operating system

- Verified by a standalone C/C++ program implemented by RACER, that has the advantage of avoiding the interference of mapping/assembling programs

Table of Contents

- Introduction
- Problem Definition
- Objective
- Related Work
- First Proposal and Defects
- H-RACER Proposal
- Evaluation
- **Conclusion**
- Future Work

Conclusion

- ➔ H-RACER acquires the major advantages of RACER in both aspects performance and time
- ➔ H-RACER added its elegant algorithm in detecting the errors types and properly applying their corrections

Conclusion

- ➔ H-RACER is the fastest with the highest accuracy algorithm among the algorithms that corrects all types of errors

- ➔ H-RACER algorithm is an open source program implemented in C/C++

Table of Contents

- Introduction
- Problem Definition
- Objective
- Related Work
- First Proposal and Defects
- H-RACER Proposal
- Evaluation
- Conclusion
- **Future Work**

Future Work

- ➔ Enhancing the memory usage of H-RACER for long genomes, so as to be able to run long genomes within 244GiB RAM

- ➔ Implementing H-RACER with parallel threads, where both time and memory will be enhanced, especially for long genomes

Thank you!