Algorithm 1 Loading and Preprocessing

```
procedure LOAD_FASTA_FILES(folder)
    list all ".fasta"/".fas" files in folder
    initialize empty dict $sequences\_dict$
    for each file in files do
        parse sequences into list
        $sequences\_dict$[file] $\leftarrow$ list of sequences
    end for
    return $sequences\_dict$
end procedure
procedure ADJUST_THRESHOLD(sequences, init_threshold)
    $n \leftarrow$ number of $sequences$
    if $n < 2$ then
        return init_threshold
    end if
    compute $avg\_sim =$ mean pairwise similarity
    $thr \leftarrow init\_threshold \times (1 + avg\_sim)$
    $thr \leftarrow \min(\max(thr, 0.2), 0.8)$
    return $thr$
end procedure
procedure PROCESS_GAPS(sequences, gap_threshold)
    convert $sequences$ to char matrix
    compute gap fraction per column
    delete columns where fraction > gap_threshold
    fill remaining gaps with most common base
    return original_sequences, processed_sequences
end procedure
```

Algorithm 2 Conserved Region Detection

```
procedure FIND_CONSERVED_REGIONS(proc_seqs, min_len, max_len, thr)
    pad all seqs to equal length and transpose
    for each column compute base frequency and label (freq > thr)
    scale features; train linear SVM; predict labels
    extract contiguous regions where label==1
    filter regions with length in [min_len, max_len]
    return regions, predictions, avg_accuracy
end procedure
procedure EXPORT_RESULTS(regions, out_folder, thr, preds, fname, acc, orig_seqs)
    create output file path
    for each region in regions do
        write header with index, fname, thr, original start, validation
        write region sequence
    end for
    write "Average accuracy: "  acc
    return output_file_path
end procedure
```

Algorithm 3 Parsing and Difference Calculation

```
1: procedure PARSE_FASTA(lines)
2:     initialize empty list seqs
3:     for each line in lines do
4:         if line starts with "¿" then
5:             start new sequence entry
6:         else
7:             append to current sequence
8:         end if
9:     end for
10:    return list of (ID, sequence)
11: end procedure
12: procedure MODIFY_AND_WRITE_SEQUENCES(test_seqs, cons_seqs, out_folder, in_file)
13:    compute gap indices from cons_seqs
14:    for each test_seq in test_seqs do
15:        remove chars at gap indices
16:        write to new file in out_folder
17:    end for
18:    return list of output file paths
19: end procedure
20: procedure CALCULATE_DIFFERENCE(test_seq, species_seqs)
21:    filter species_seqs length $\geq$ len(test_seq)
22:    compute mismatches and p_obs per sequence
23:    $avg\_diff \leftarrow \overline{p\_obs} \times 100$
24:    build pct_distribution and base_distribution
25:    return $avg\_diff$, pct_distribution, base_distribution
26: end procedure
27: procedure PROCESS_DIFFERENCE(test_file, species_folder, out_xlsx)
28:    read test segments
29:    for each segment do
30:        call CALCULATE_DIFFERENCE
31:        collect summary, pct_details, base_details
32:    end for
33:    write DataFrames to Excel
34:    generate heatmaps for pct and base details
35: end procedure
```

Algorithm 4 Identity Calculation and Utilities

```
 1: procedure CALCULATE_IDENTITY(test_seq, species_seqs)
 2:     filter species_seqs length ≥ len(test_seq)
 3:     compute single_base identity rates
 4:     build consensus sequence
 5:     compute avg_percent_identity and percent_distribution
 6:     return avg_percent_identity, percent_distribution, single_base_rates
 7: end procedure
 8: procedure PROCESS_IDENTITY(test_file, species_folder, out_xlsx)
 9:     same as PROCESS_DIFFERENCE but use CALCULATE_IDENTITY
10: end procedure
11: procedure READ_ACCESSION_NUMBERS(txt_file)
12:     read each nonempty line as accession
13:     return list of accession numbers
14: end procedure
15: procedure FETCH_GENOME_FROM_ACCESSION(acc)
16:     call Entrez.efetch for FASTA
17:     return SeqRecord or None
18: end procedure
19: procedure FETCH_CDS_FROM_ACCESSION(acc)
20:     call Entrez.efetch for GenBank
21:     extract features with type "CDS"
22:     return list of CDS features
23: end procedure
24: procedure PROCESS_FASTA_FILES(folder, modify_titles, new_title, re-
    name_flag)
25:     list .fasta/.fas files
26:     for each file do
27:         clean degenerate bases; modify title per flags
28:         overwrite file
29:     end for
30: end procedure
31: procedure PROCESS_FASTA_FOR_PRIMERS(in_file, out_xlsx, primer_len)
32:     parse FASTA into name→seq dict
33:     for each seq do
34:         extract forward/reverse primers
35:         compute GC% and Tm
36:     end for
37:     write results to Excel
38: end procedure
39: procedure REPLACE_BASES_AND_FORMAT(in_file, out_docx)
40:     read text from txt or docx
41:     split into chunks of max_capacity
42:     for each chunk do
43:         generate QR code with rounded modules and gradient
44:         compress and save image
45:     end for
46:     create Word doc; replace bases with styled runs
47:     save doc; return image paths
48: end procedure
49: procedure GENERATE_HEATMAP(data, title, prefix, annotate)
50:     choose rasterization if large
51:     call seaborn.heatmap with vmin=0, vmax=100
52:     customize ticks for num and prop heatmaps
53:     save PNG and SVG
54: end procedure
```