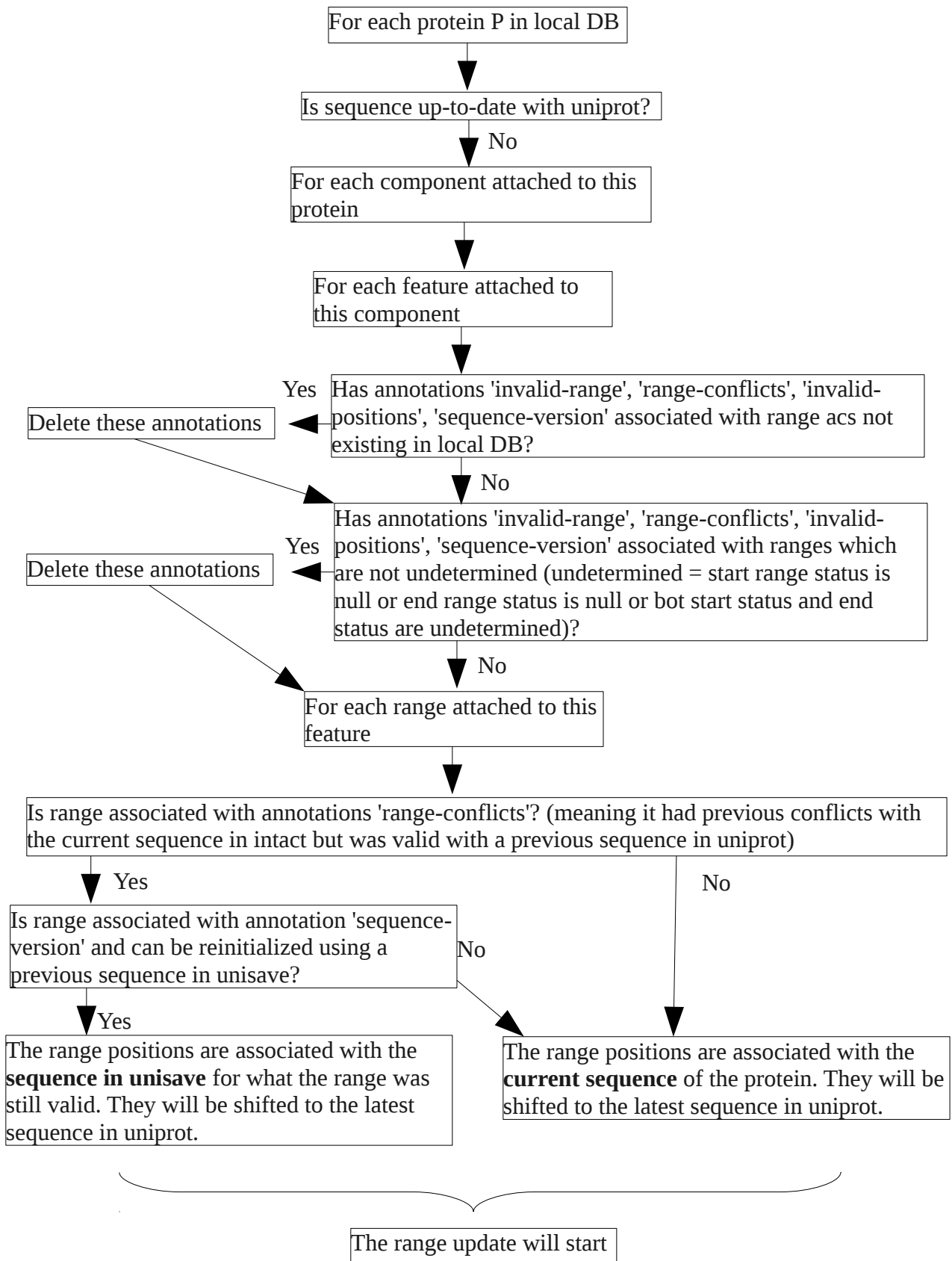
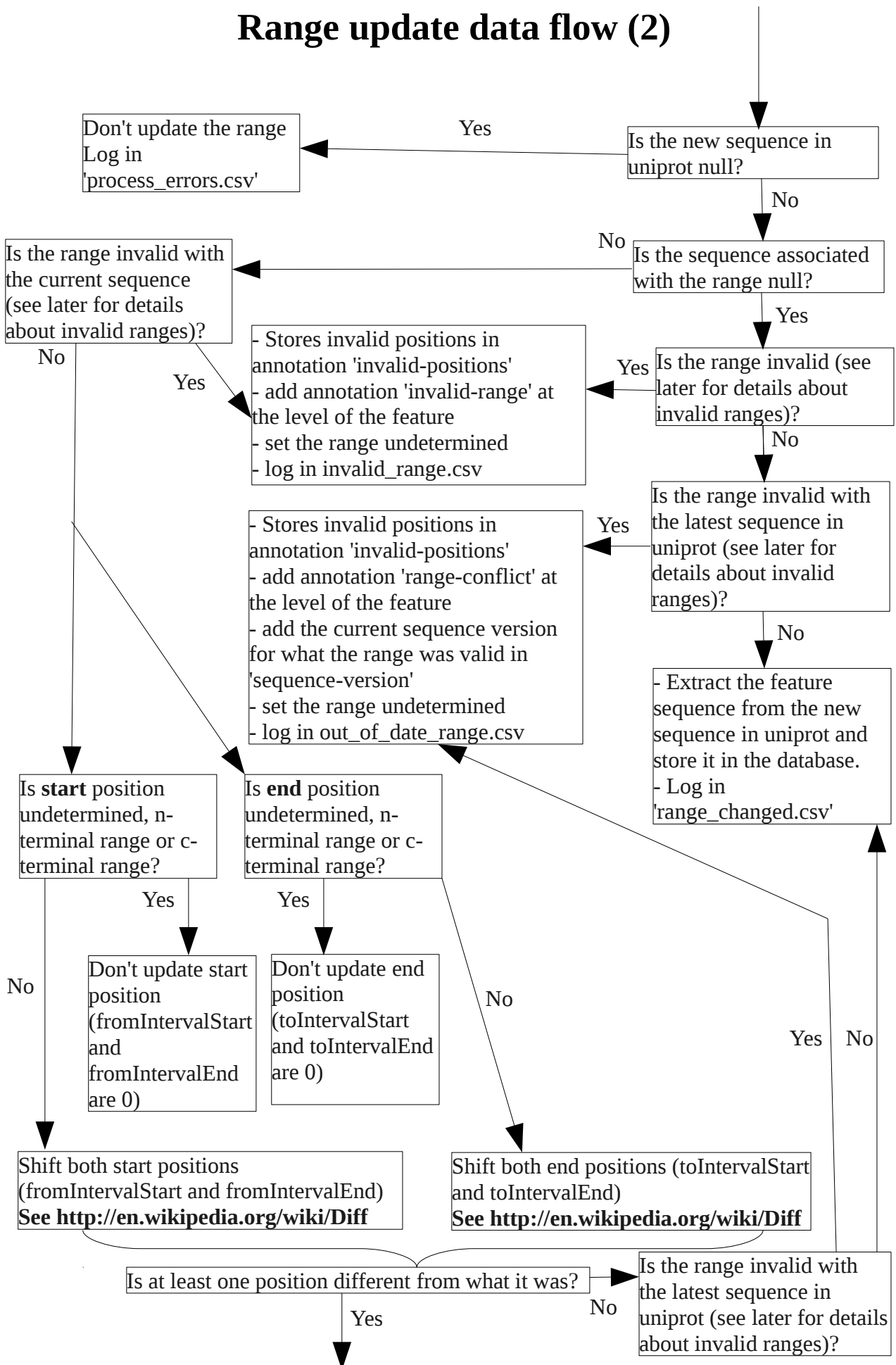


Range update data flow



Range update data flow (2)



Range update data flow (2)

Deduce the first position of the range by subtracting end position with the previous feature sequence length.

The current algorithm is not dealing with insertion of a different amino acid just before the feature sequence when the insertion occurs exactly at the same position as the start position of the feature was before.

The direct consequence of such a case is that the end of the feature can be shifted properly but the start position of the feature is undetermined (which doesn't mean that the feature doesn't exist anymore!).

- Stores invalid positions in annotation 'invalid-positions'
- add annotation 'range-conflict' at the level of the feature
- add the current sequence version for what the range was valid in 'sequence-version'
- set the range undetermined
- log in out_of_date_range.csv

Yes

Is start equal to 0 and end strictly superior to 0?

No

Is range invalid with shifted positions and new uniprot sequence?

Yes

Extract the feature sequence using the shifted positions and the new uniprot sequence

No

Is feature sequence conserved?

No

Yes

- Update the range positions
- log in 'range_changed.csv'

Yes

- Update the range positions with the corrected shifted positions
- log in 'range_changed.csv'

No

Is the previous feature sequence a sub-sequence of the new feature sequence? (the algorithm is not dealing with insertions of one to several amino acid(s) upstream the feature if the inserted amino acid(s) is(are) identical to the first amino acid of the feature. The direct consequence is that the new feature sequence contains all the inserted amino acids and we have to correct the shifted position for the first position of the feature)

Yes

Add a caution at the level of the feature

Was the feature n-terminal or c-terminal and is not anymore?

