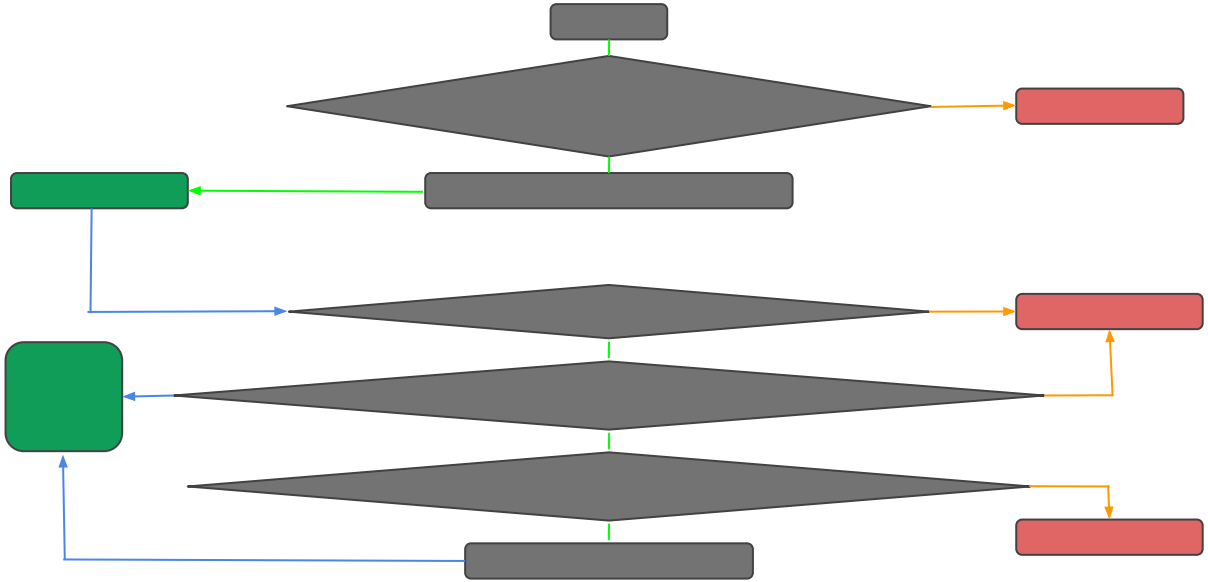
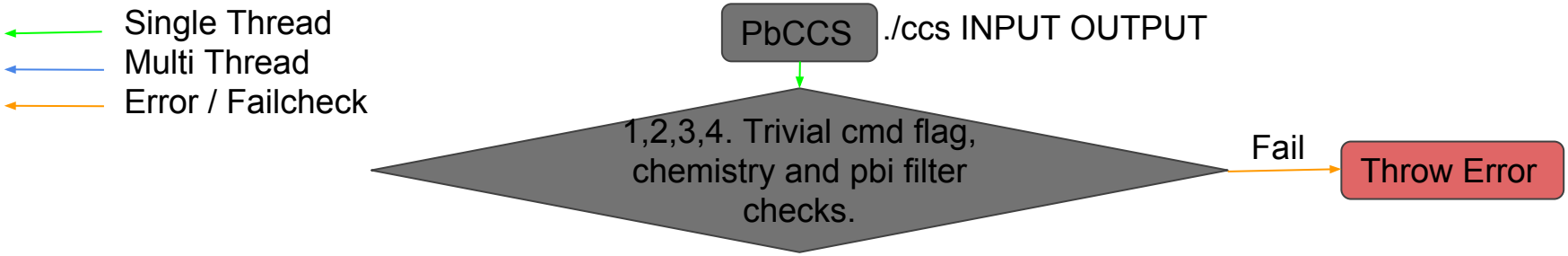
Unboxing CCS2

-- Avi Srivastava

Single Thread
Multi Thread
Error / Failcheck

Flowchart

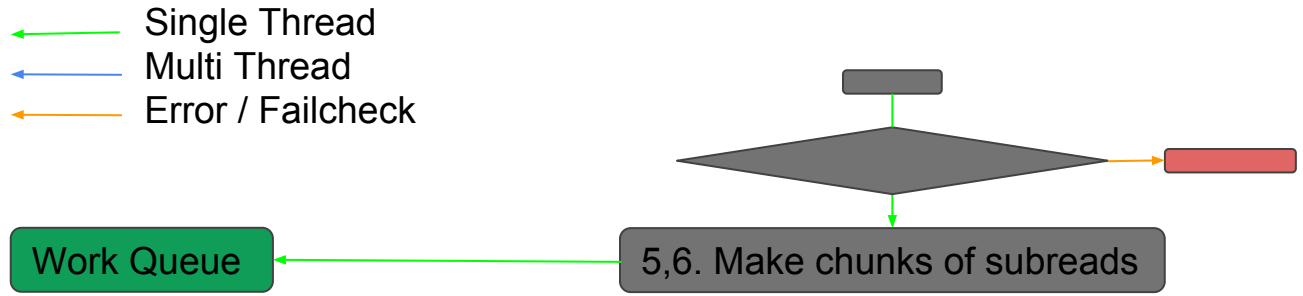




Check for command line arguments:

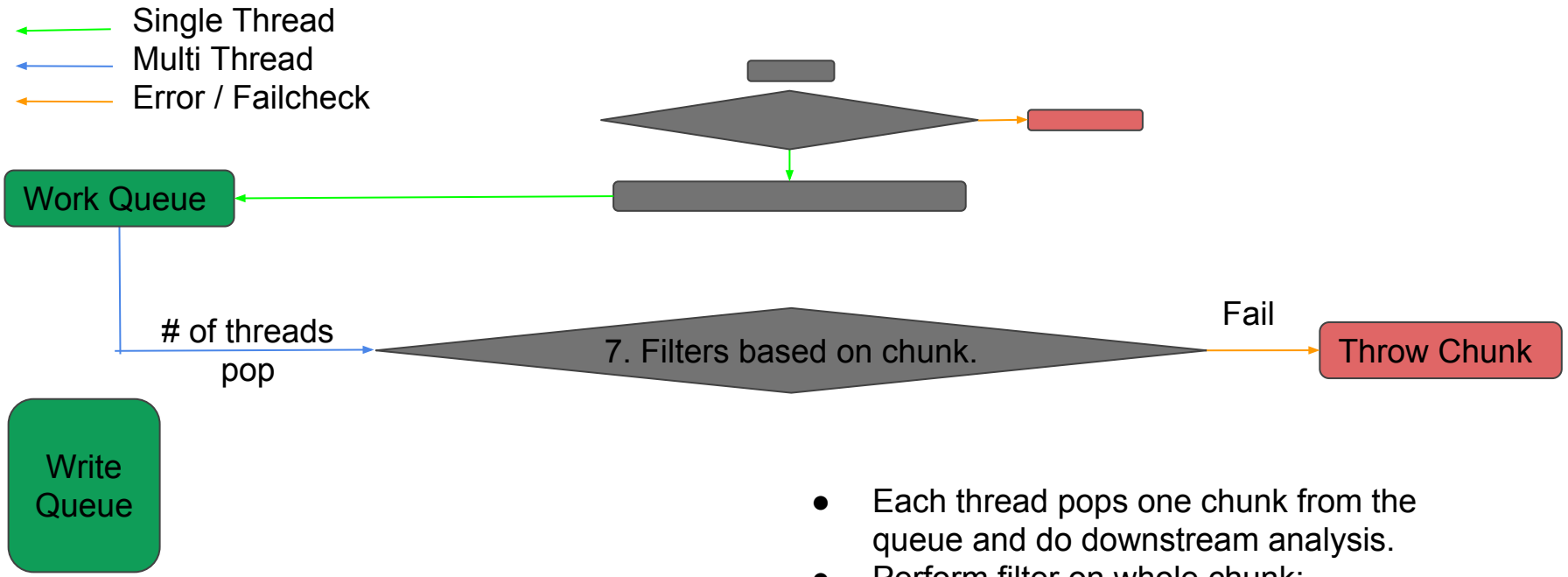
- minPasses >= 1
- Input / Output format
- Too many arguments
- Output file already exist
- Option --byStrand not compatible with --noPolish and similar-ish…

Directly throws Error without doing anything.

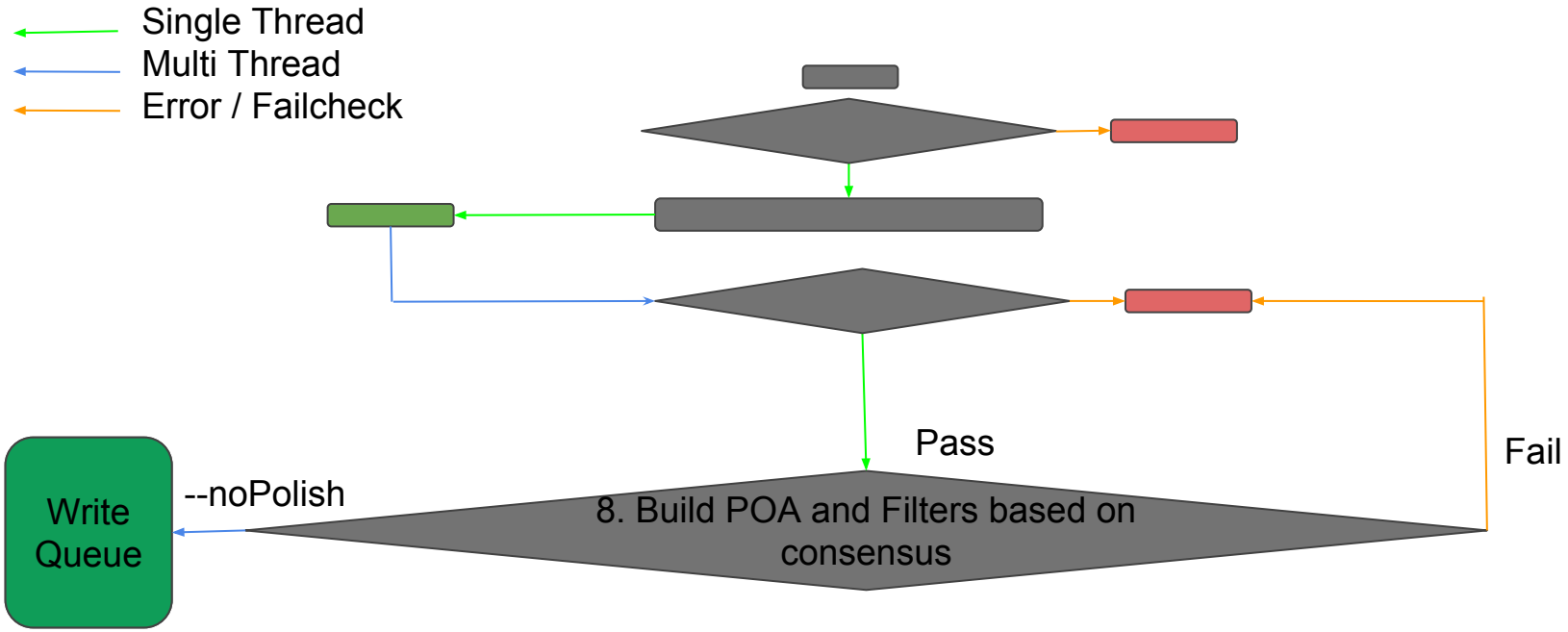


Write Queue

- Initiate work / write queue based on # of threads.
- Check required output format and initialize output stream accordingly (can be FASTQ / BAM).
- Subsample all the subreads from one ZMW and push them in worker queue.

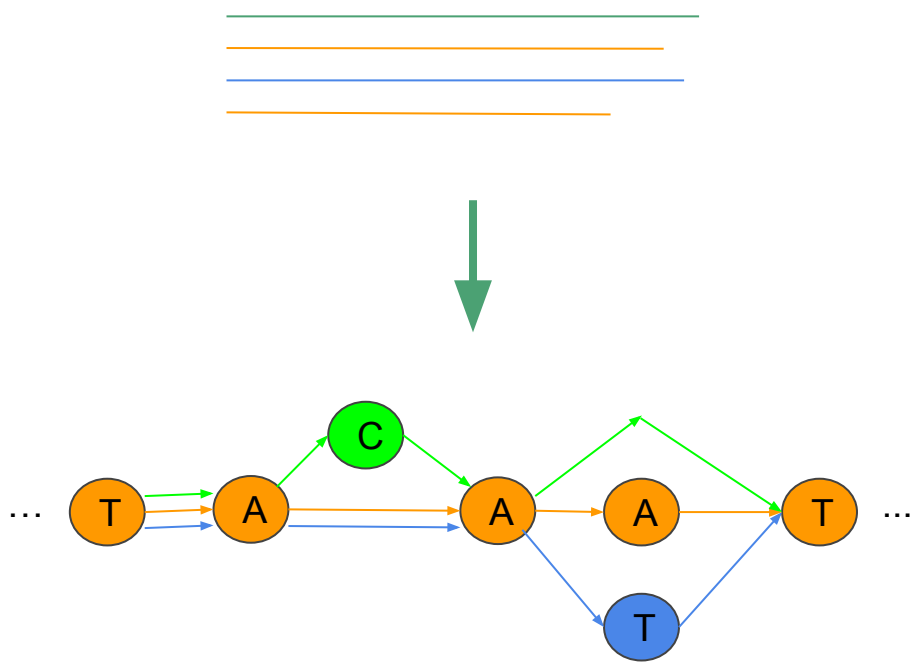


- Each thread pops one chunk from the queue and do downstream analysis.
- Perform filter on whole chunk:
  - Min SNR(A,C,T,G) > SNR thresh.
  - Median Read Length < 10
  - # Passes < Pass thresh.
- Perform filter on each read:
  - Read quality < 0.65
  - size(Read) > 2* Median read len.
- Sort remaining reads based on deviation from median and build POA.

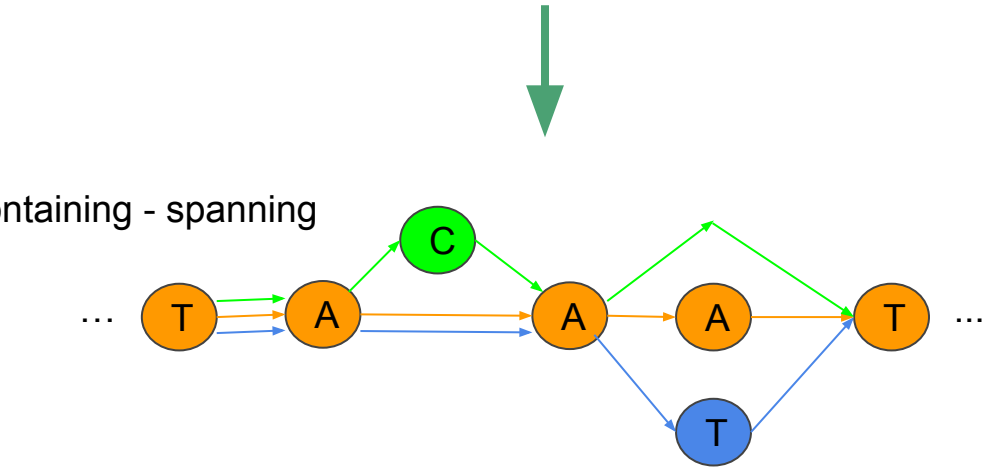

Single Thread
Multi Thread
Error / Failcheck
Pass
Fail
--noPolish
Write Queue
8. Build POA and Filters based on consensus

POA (Partial Order Alignment)

POA (Partial Order Alignment)



POA (Partial Order Alignment)



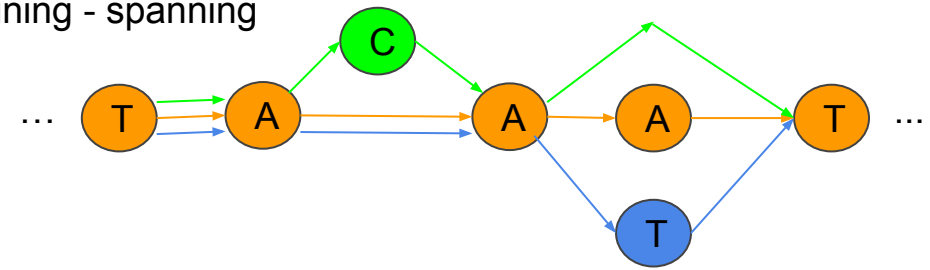
Scoring: 2*containing - spanning

Score: (2*3-3)+(2*2-3) + (2*1-3) +(2*1-1) + ...

POA (Partial Order Alignment)

Subreads

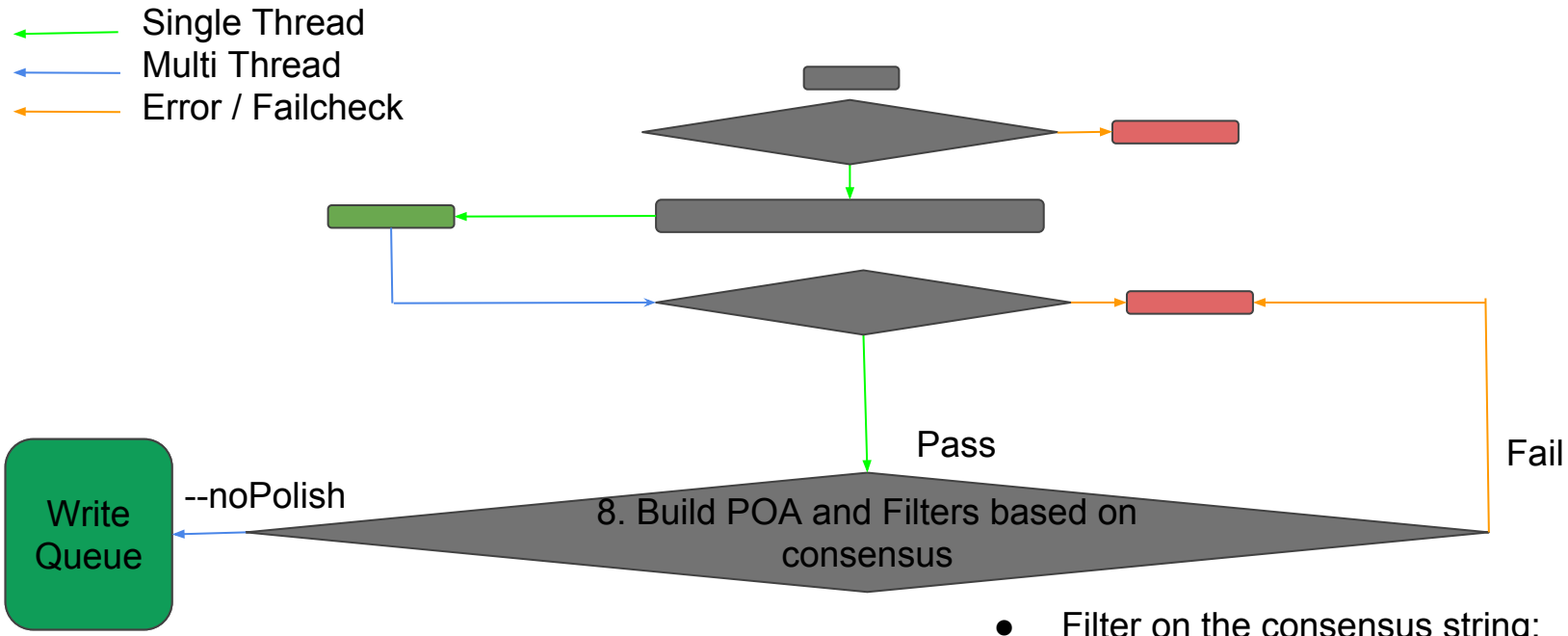
Scoring: 2*containing - spanning



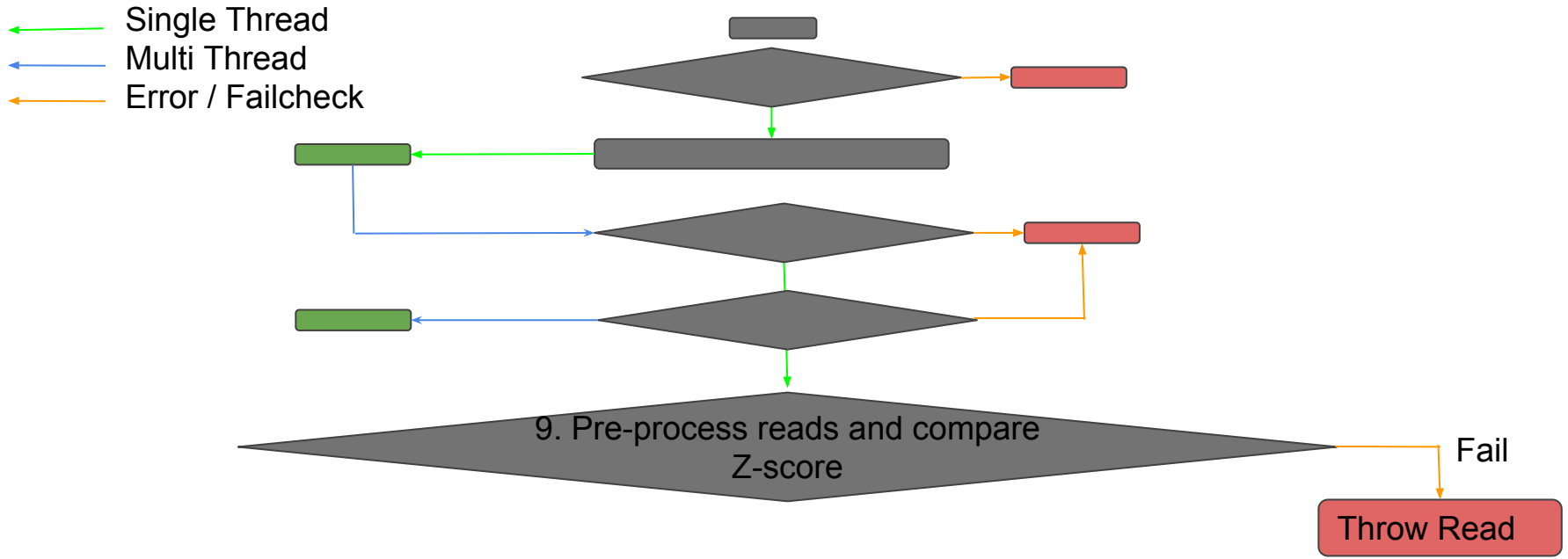
Score: (2*3-3)+(2*1-3) + (2*1-3) + ...

Score: (2*3-3)+(2*2-3) + (2*1-3) +(2*1-1) + ...

Score: (2*3-3)+(2*2-3) + (2*1-3) +(2*1-1) + ...



- Filter on the consensus string:
  - Throw chunk if consensus is too short or long.**
- Based on --noPolish push the consensus to write queue and exit.
- Otherwise go to next stage of Polishing.

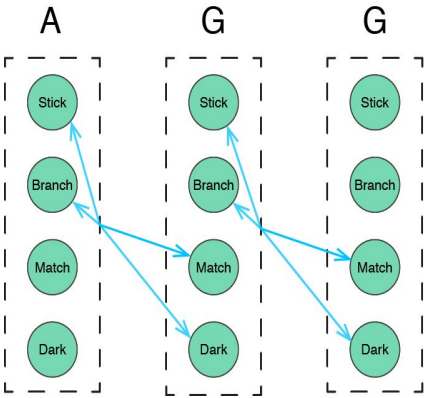


- **Zscore:** How likely is the probability of; the read must be coming from the consensus sequence under pacbio model.

Z-Score

P(T / R) : Posterior probability of template given a set of reads.

We have to maximize this posteriori for an unknown template given a set of subreads.



Bayes: $P(T / R) = P(R / T) * P(T) / P(R)$

To maximize P(T / R) we have to maximize P(R / T)

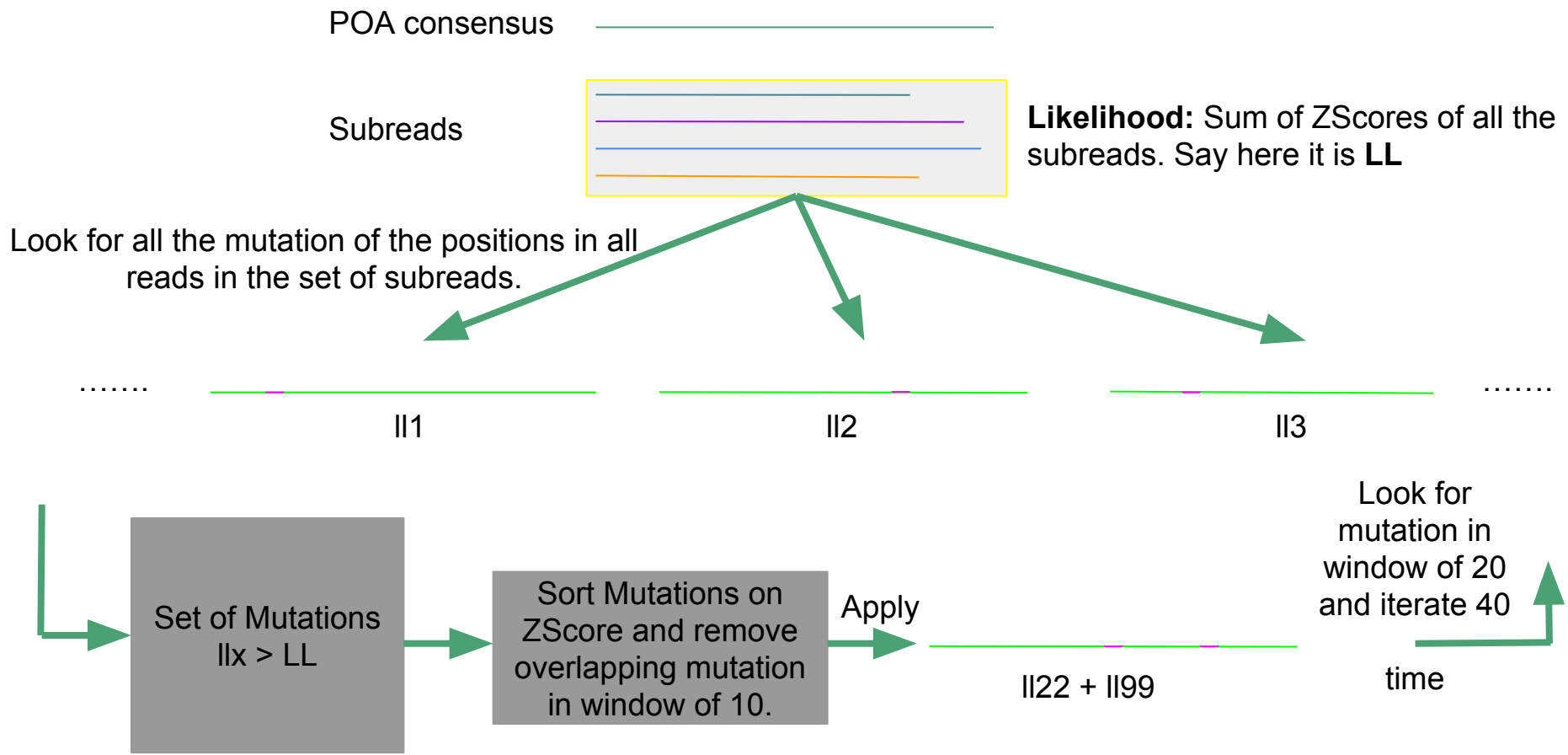
$P(R / T) = \text{Product}(P(R_i / T))$ where R_i is a read from set R

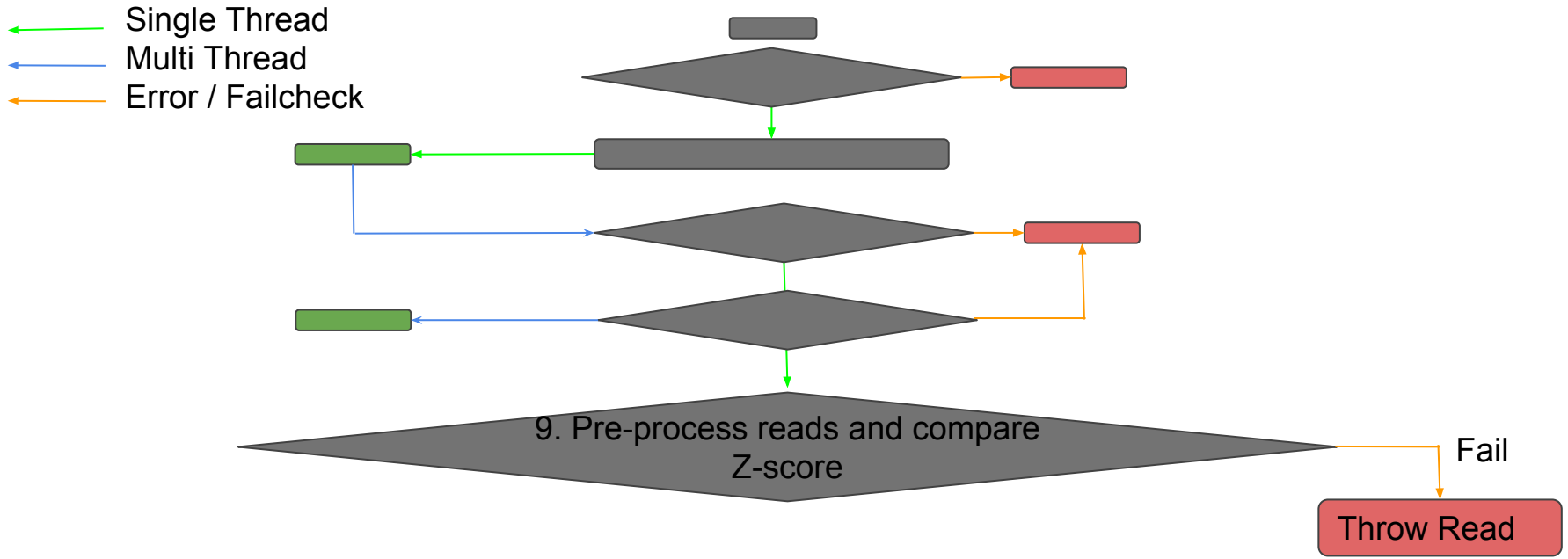
- $P(R_i / T) = \text{Sum}(P(R_i, A / T))$
  Where A represents all the variations of the read

Instead of Summing we can just look for most probable alignment A for each read using Forward Backward Algorithm.

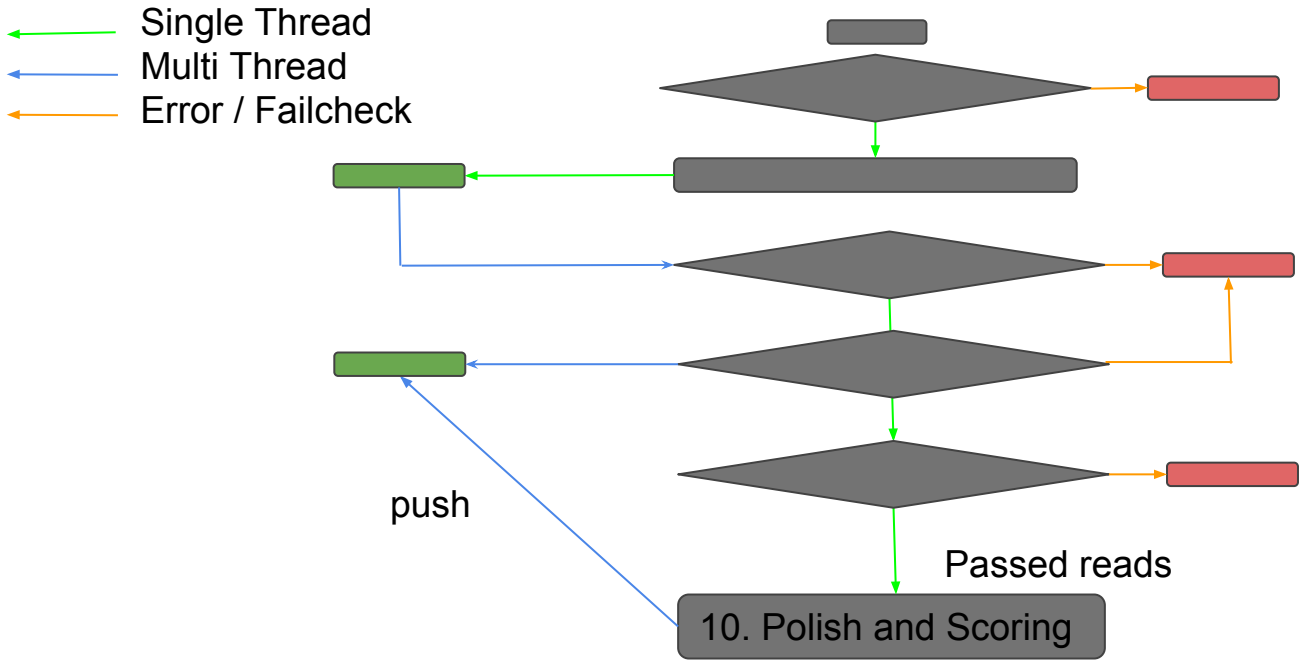
Based on the chemistry we have pre-learned emission and transition probabilities.

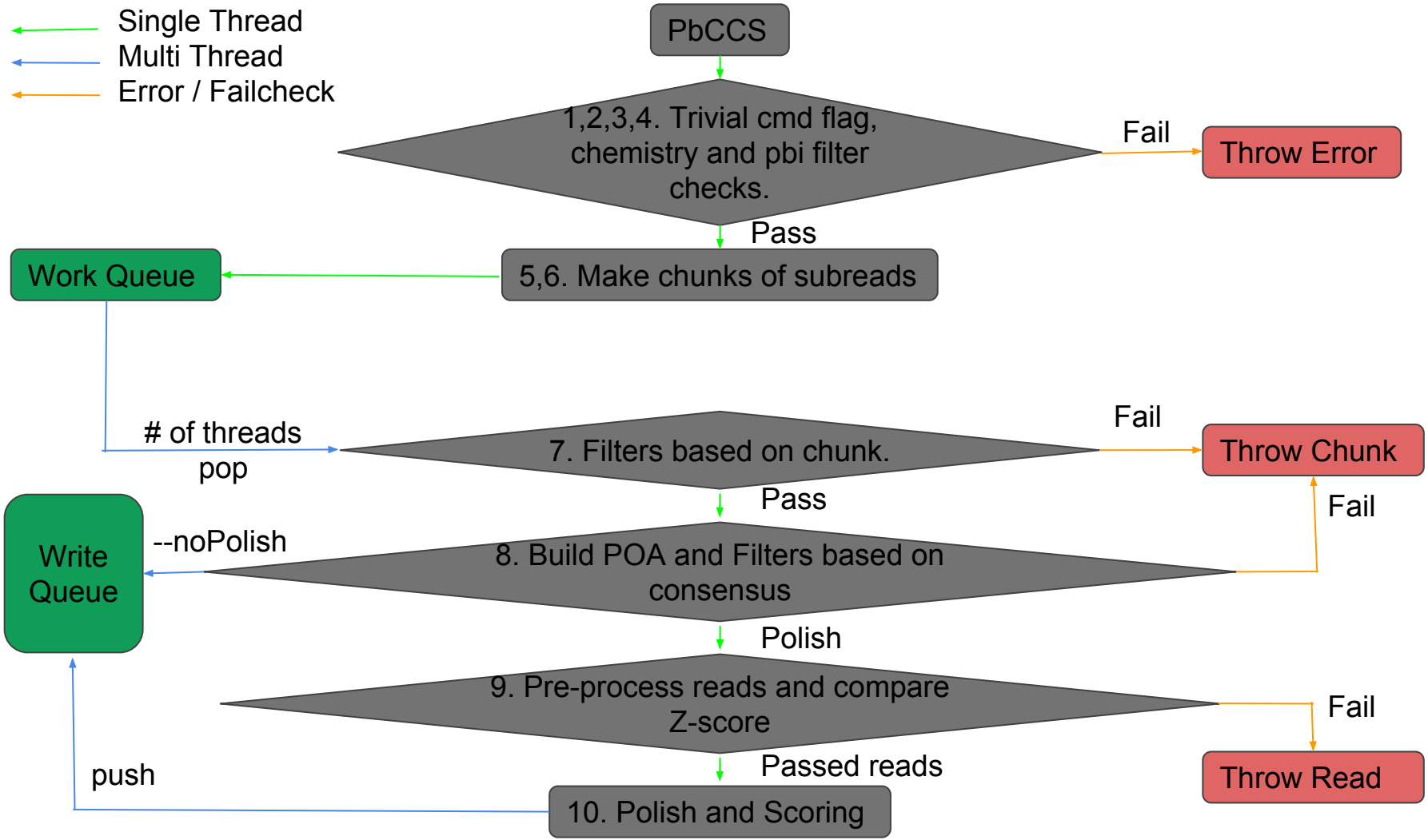
Polish


POA consensus
Subreads
Likelihood: Sum of ZScores of all the subreads. Say here it is LL
Look for all the mutation of the positions in all reads in the set of subreads.
.....
ll1
ll2
ll3
.....
Set of Mutations llx > LL
Sort Mutations on ZScore and remove overlapping mutation in window of 10.
Apply
ll22 + ll99
Look for mutation in window of 20 and iterate 40
time



- **Zscore:** How likely is the probability of; the read must be coming from the consensus sequence under pacbio model.
- Throw read:
  - Zscore < 12.5
- Throw chunk:
  - # passes < thresh. Passes


Single Thread
Multi Thread
Error / Failcheck
push
Passed reads
10. Polish and Scoring


Single Thread
Multi Thread
Error / Failcheck
PbCCS
1,2,3,4. Trivial cmd flag, chemistry and pbi filter checks.
Fail
Throw Error
Pass
Work Queue
5,6. Make chunks of subreads
# of threads
pop
7. Filters based on chunk.
Fail
Throw Chunk
Pass
Fail
Write Queue
--noPolish
8. Build POA and Filters based on consensus
Polish
9. Pre-process reads and compare Z-score
Fail
Throw Read
Passed reads
push
10. Polish and Scoring