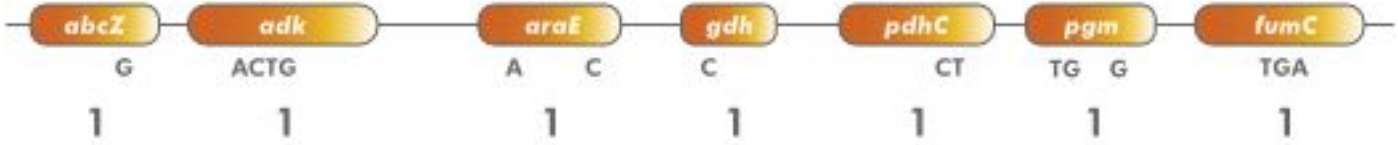
chewBBACA

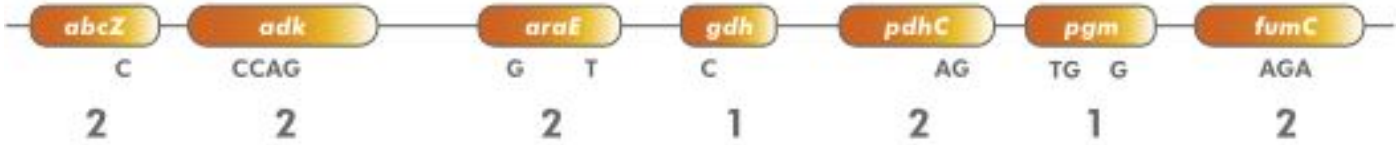
IGC 2018
Mickael Silva

7 housekeeping genes

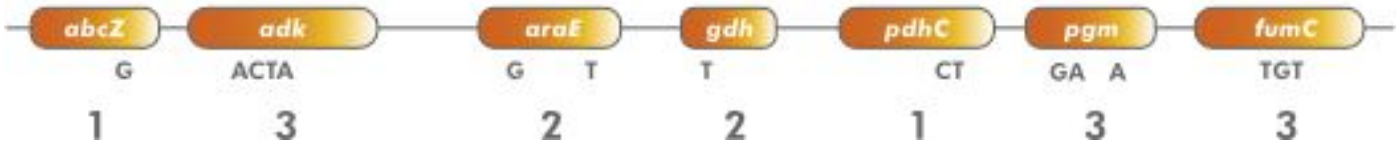
Strain A



Strain B



Strain C



	Locus 1	Locus 2	Locus 3	Locus 4	Locus 5	Locus 6	ST
Strain A	1	1	1	1	1	1	1
Strain B	2	2	1	2	2	2	10
Strain C	1	3	2	2	1	3	20

Task 1 - Organizing and fetching the data

- Get chewBBACA
- Create Folders
- Download necessary genomes

Create Schema

Traditional MLST schemas relied in 7 loci that were internal fragments of housekeeping genes and each locus was defined by its amplification by a pair of primers yielding a fragment of a defined size.

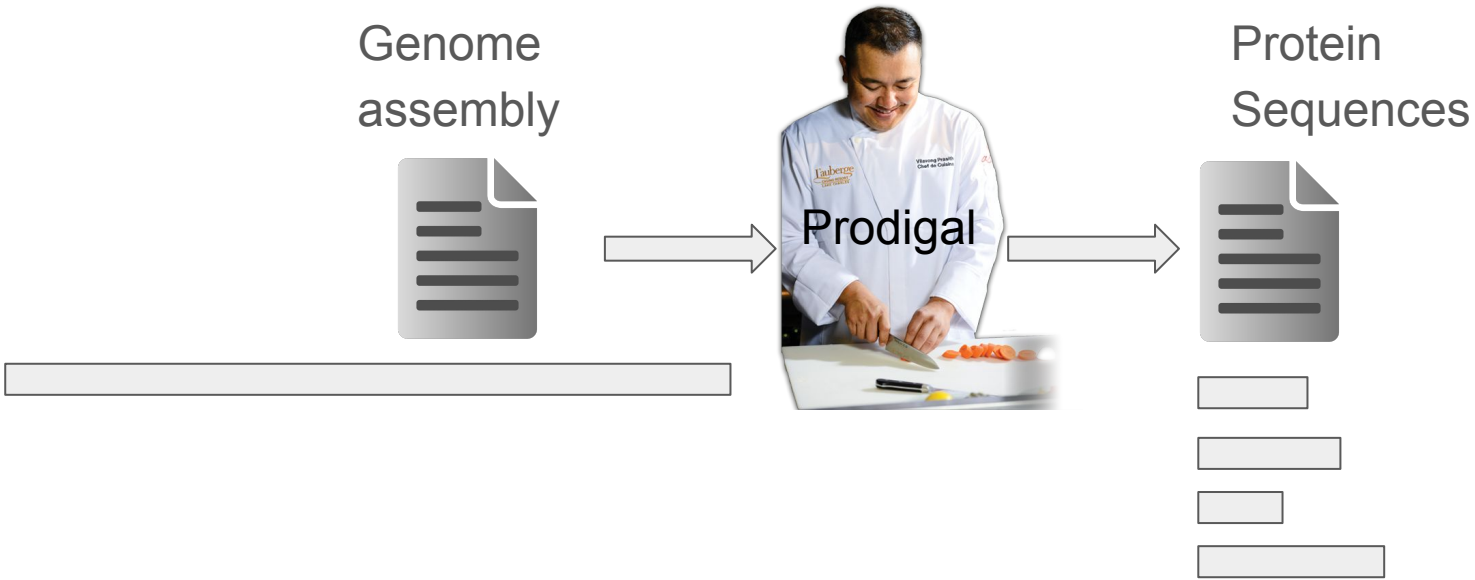
In chewBBACA, schemas are composed of **loci defined by CDSs** and all the called **alleles of a given locus are CDSs as defined by Prodigal software**. The use of Prodigal, instead of simply ensuring the presence of start and stop codons, **adds an extra layer of confidence in identifying the most probable CDS** for each allele.

Task 2 - Create a schema based on the downloaded complete genomes

chewBBACA.py CreateSchema -i complete_genomes/ -o schema_seed --cpu 6 --ptf /home/participant/oneida_tools/prodigal_training_files/Streptococcus_agalactiae.trn

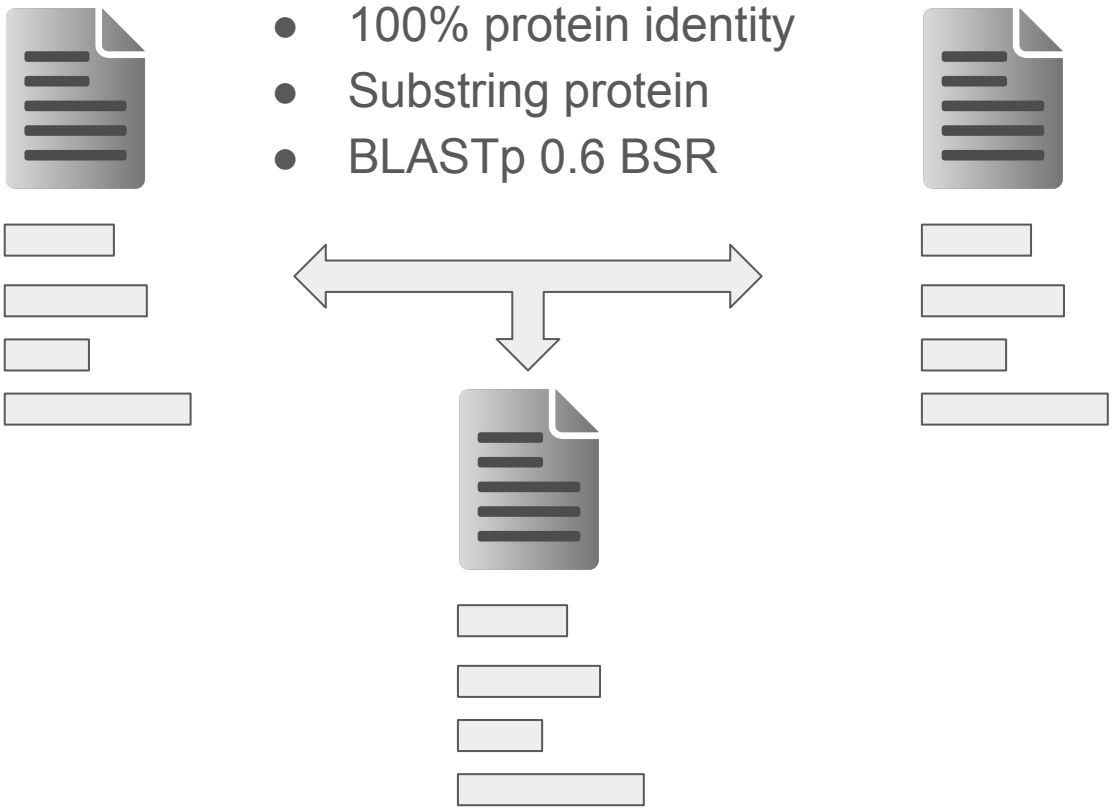
How does it work?

Create Schema - getting the Coding Sequences (CDS)

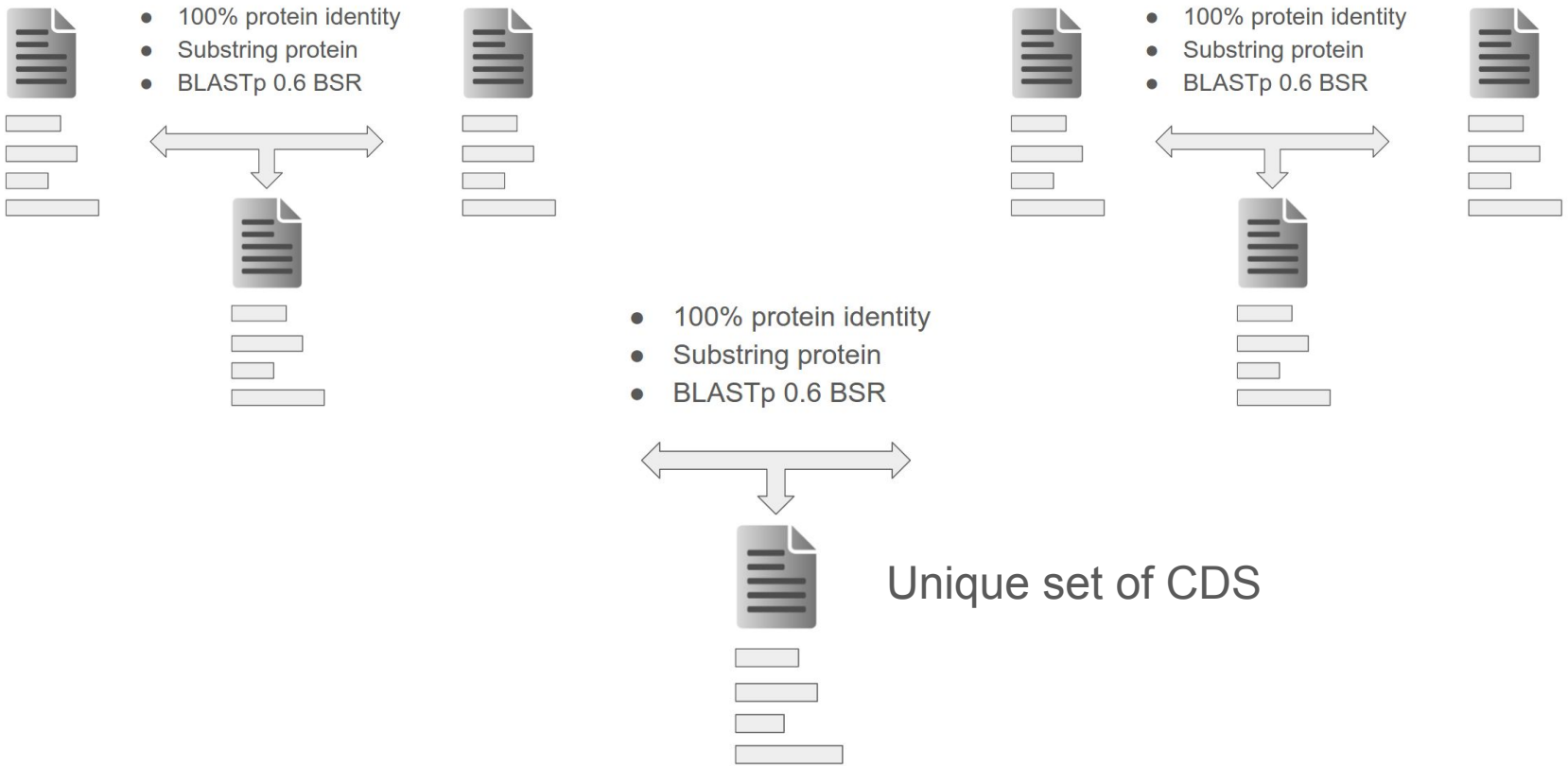


Prodigal - (http://prodigal.ornl.gov/) (Hyatt D et al BMC Bioinformatics 2010)

Create Schema - getting the unique CDS between two sets

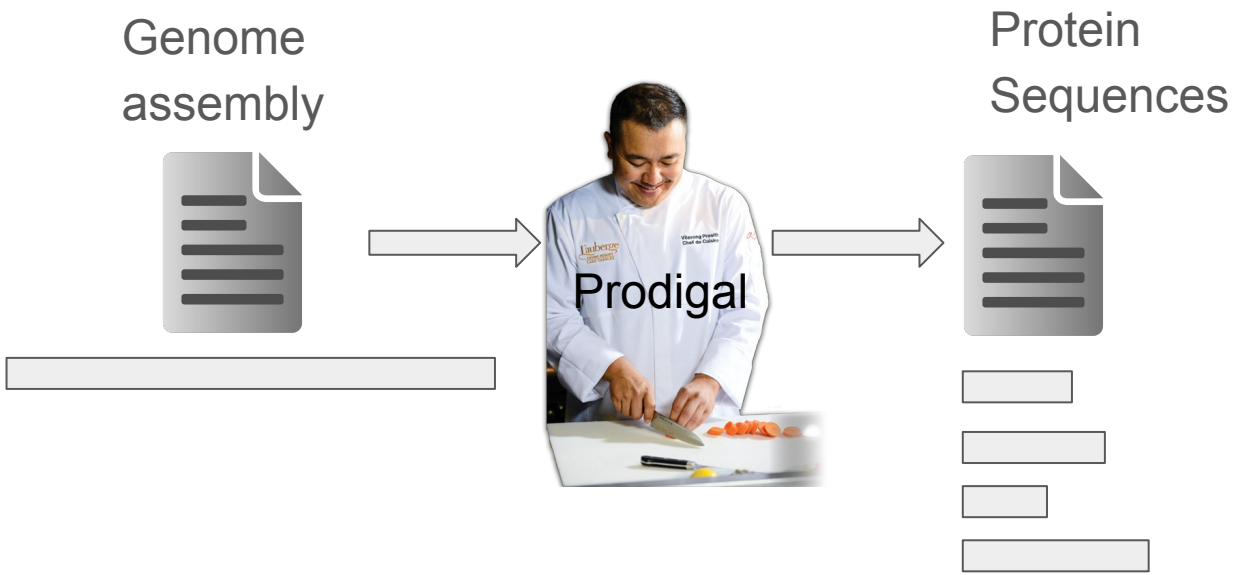


Create Schema - pairwise comparison until unique set of CDS

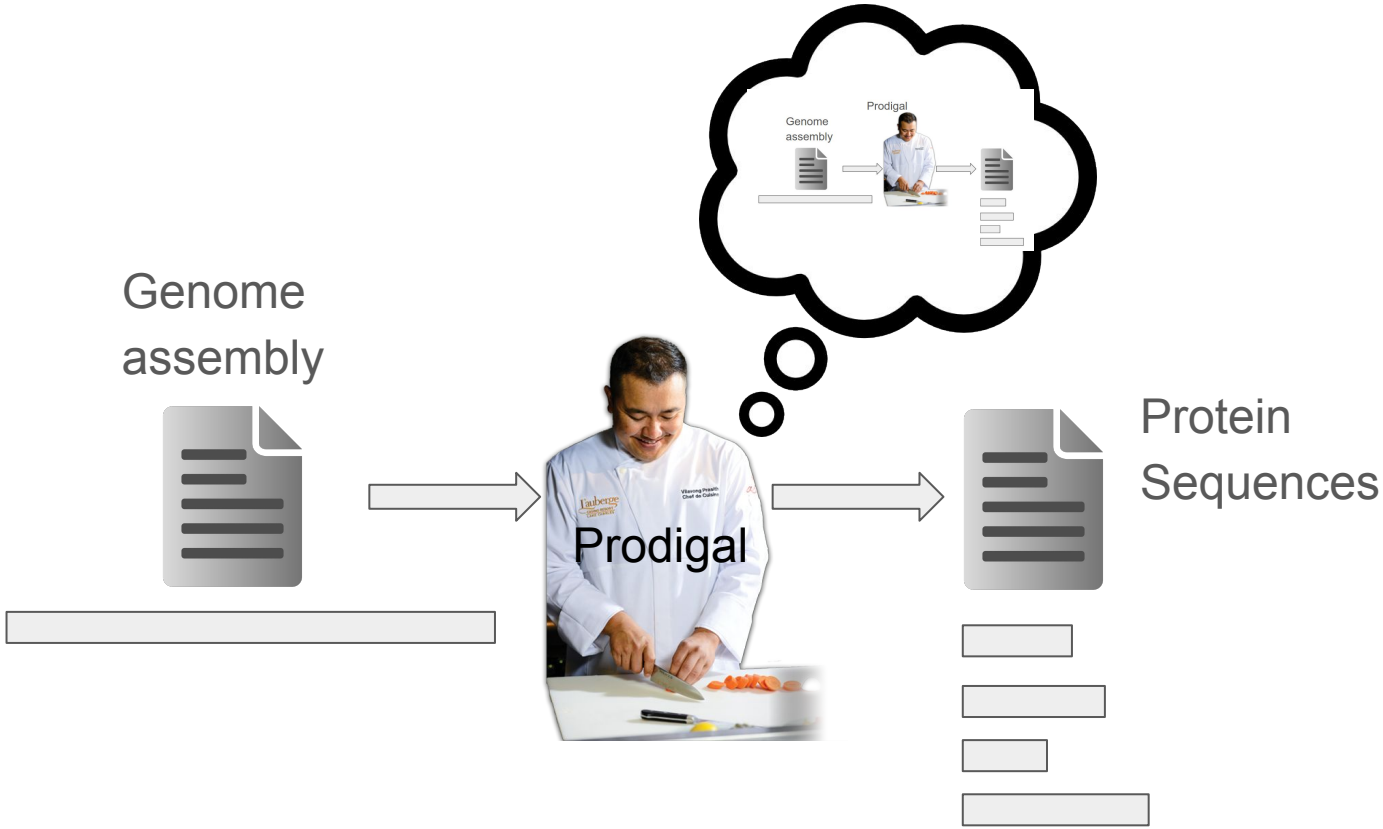


Create Schema - getting the Coding Sequences (CDS)

chewBBACA.py CreateSchema -i complete_genomes/ -o schema_seed --cpu 8 **--ptf /home/participant/oneida_tools/prodigal_training_files/Streptococcus_agalactiae.trn**



Create Schema - training file



ATGGCAATATTTTTCATGATTTTTCTGATTGTTTGTGTGCTCCTATTGGTGATAGTCACA
CTGAGTACAGTTTATGTGGTTCGTCAGCAGTCGGTGGCGATTATTGAACGCTTTGGG
AAATACCAAAAGGTTGCTAATAGCGGTATTCATATTCGCTTGCCTTTTGGGATTGACTC
GATTGCAGCACGGATTCAGTTGCGCTTGTTGCAAAGTGATATTGTGGTTGAGACTAA
GACCAAGGACAATGTGTTCGTTATGATGAATGTAGCGACTCAGTACCGTGTCAACGA
GCAGAGCGTGACAGATGCTTACTATAAACTCATACGTCCAGAATCTCAGATTAAATCTT
ATATCGAAGATGCTCTTCGCTCTTCTGTTCCAAAATTAACCTTGGATGAATTGTTTGAG
AAAAAAGATGAGATTGCCCTTGAAGTTCAACACCAAGTAGCAGAAGAAATGACCACTT
ACGGCTACATTATCGTGAAAACCTTGATTACCAAGGTCGAACCGGATGCAGAAGTTAA
GCAATCCATGAATGAAATCAATGCGGCGCAACGTAAGCGGGTCGCAGCACAAGAATT
GGCGGAAGCTGACAAGATTAAAATTGTCACTGCAGCTGAAGCCGAAGCAGAAAAAG
ACCGCCTTCATGGTGTGGGGATTGCCCAACAACGTAAGGCGATTGTGGATGGATTG
GCAGAGTCTATCACCGAACTCAAGGAAGCCAATGTTGGCATGACAGAAGAACAAATC
ATGTCTATCCTCTTGACCAACCAGTATTTGGATACCTTGAATACCTTTGCCTCTAAAGG
AAATCAAACCATCTTTTTACCAAATACGCCAAATGGTGTGGATGATATCCGAACACAAA
TCTTGTCAGCCCTTCGCGCTGAGAAGAAATAA

S.pneumoniae protease
http://www.ncbi.nlm.nih.gov/protein/446886697

Frame 1: Start, Stop
Frame 2: Start
Frame 3: Start, Stop

High number of initiator codons, in different frames (reversed ORF's not being taken in account)

Using your own prodigal file

prodigal -i referenceGenome.fasta -t tfName.trn -p single

chewBBACA.py CreateSchema -i genomes/ -o schema_seed --cpu 6 --ptf tfName.trn

Create Schema - output

- One fasta per unique CDS where all allele sequences will be stored when calling the alleles

- A folder “short” with a fasta per unique CDS where only the most variable alleles will be stored

- A file with the necessary information to traceback the name of each CDS to its origin (assembly, contig and position)

Create Schema - other arguments

--bsr 0.6

Minimum BSR for locus similarity.

--b /path/to/blastp

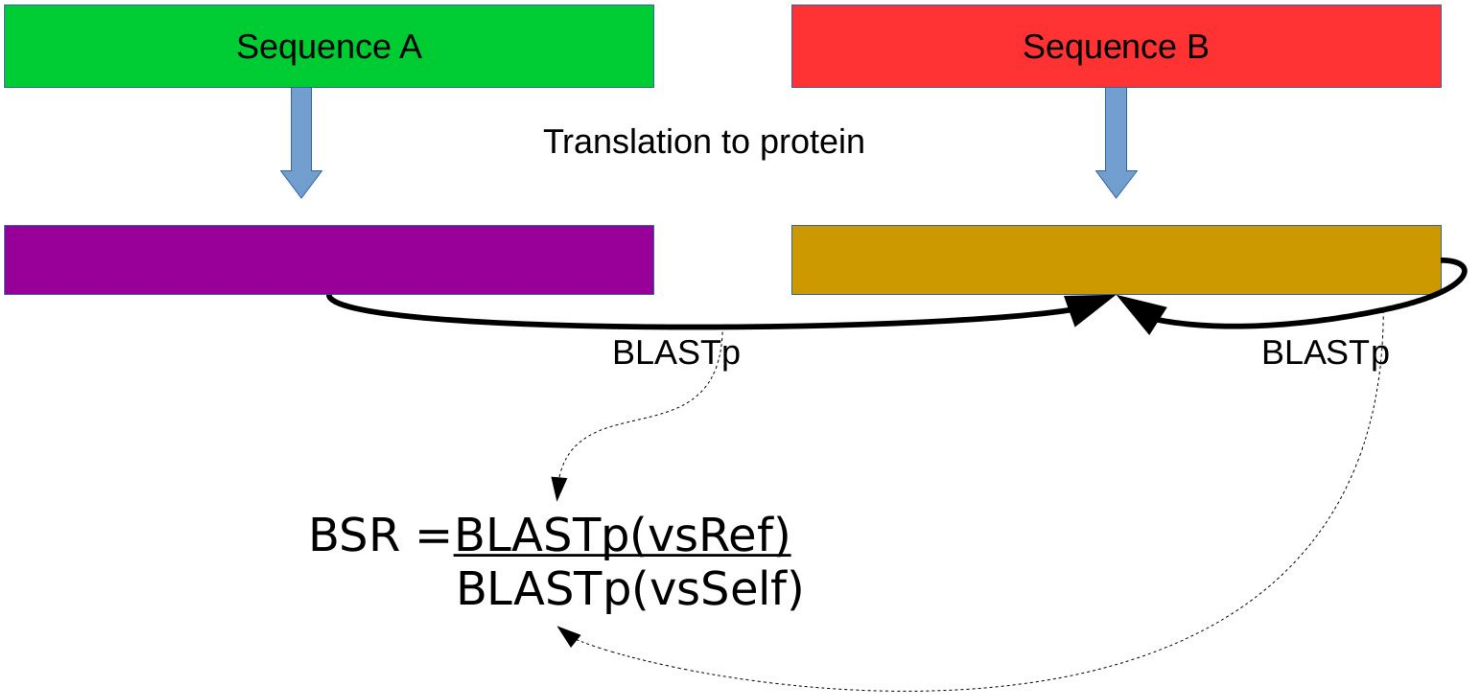
blastp full path (default: blastp)

-v --verbose

Increase output verbosity

-h, --help

BSR? BLAST SCORE RATIO



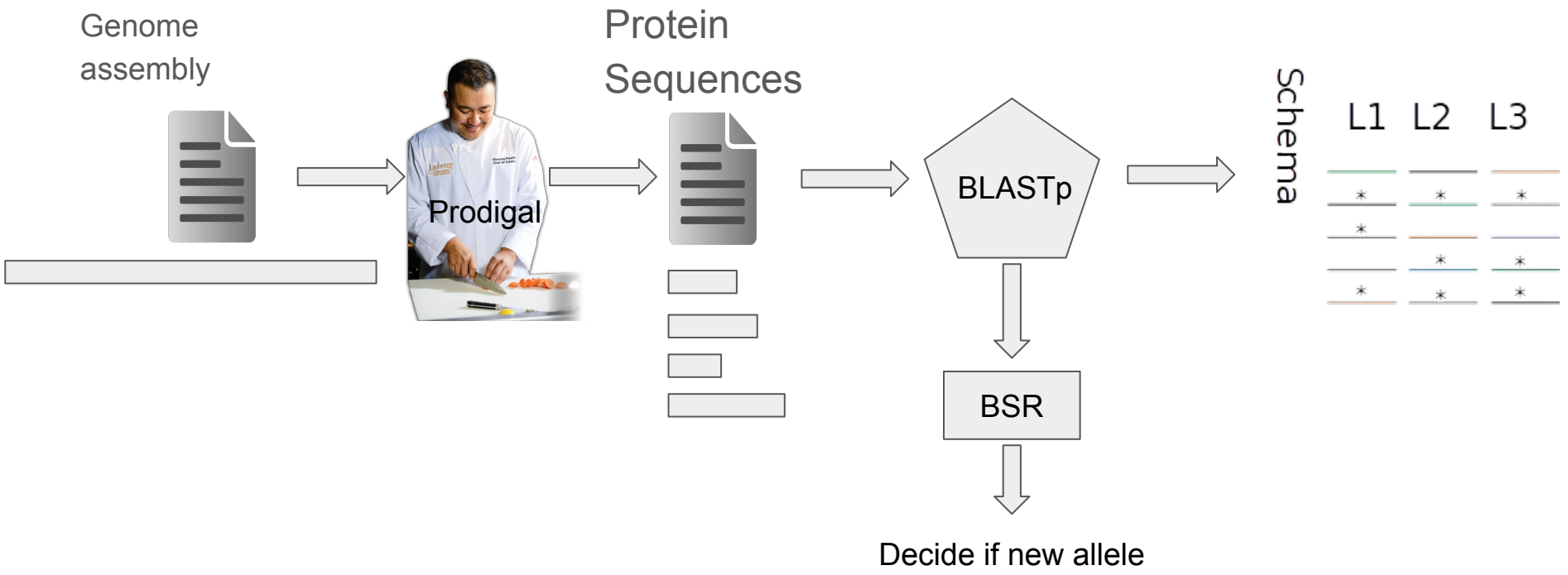
BSR – BLASTp Score Ratio (David A Rasko et al BMC Bioinformatics 2005)

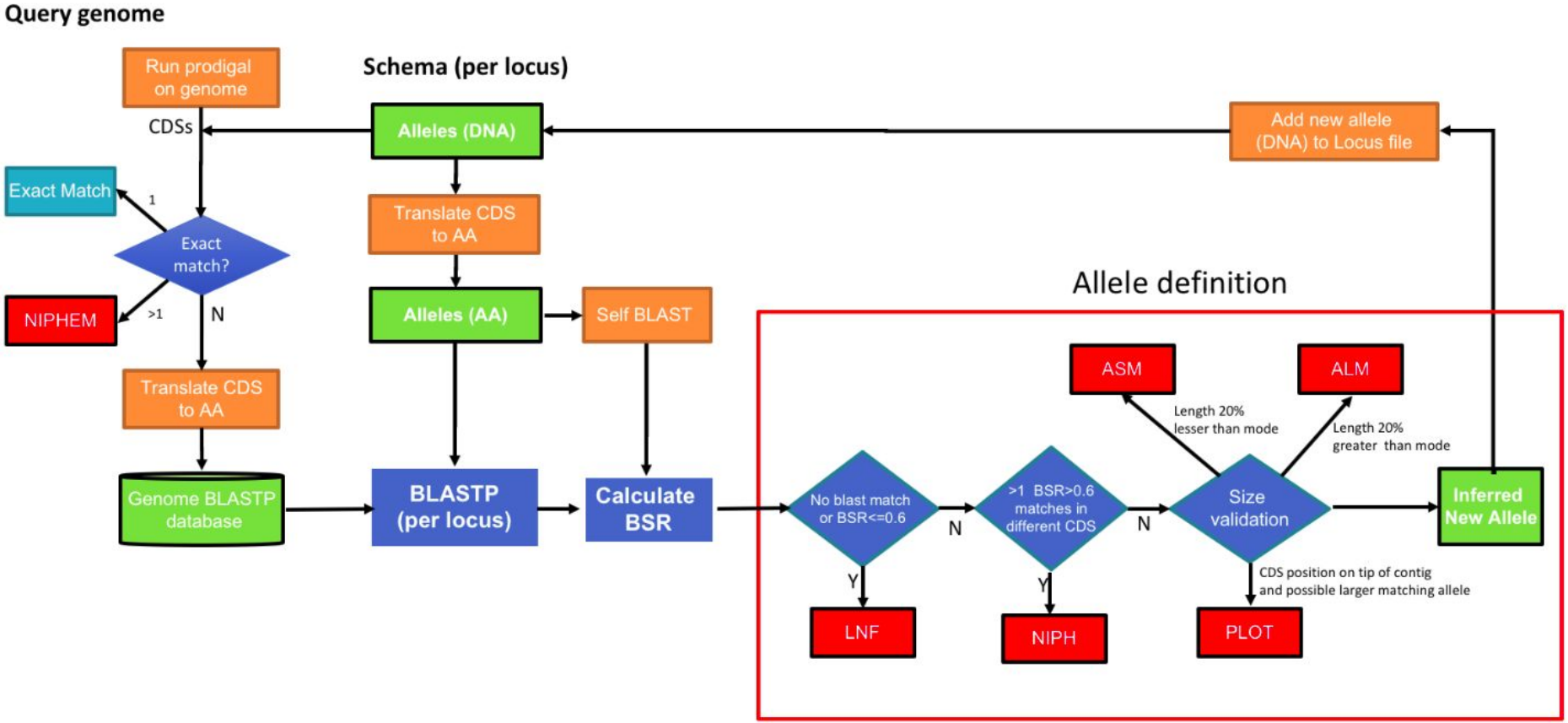
Allele Calling

Task 3 - Run an allele call using the created schema and the genomes available

chewBBACA.py AlleleCall -i listgenomes.txt -g schema_seed/ -o results --cpu 6 --ptf /home/participant/oneida_tools/prodigal_training_files/Streptococcus_agalactiae.trn

chewBBACA allele calling

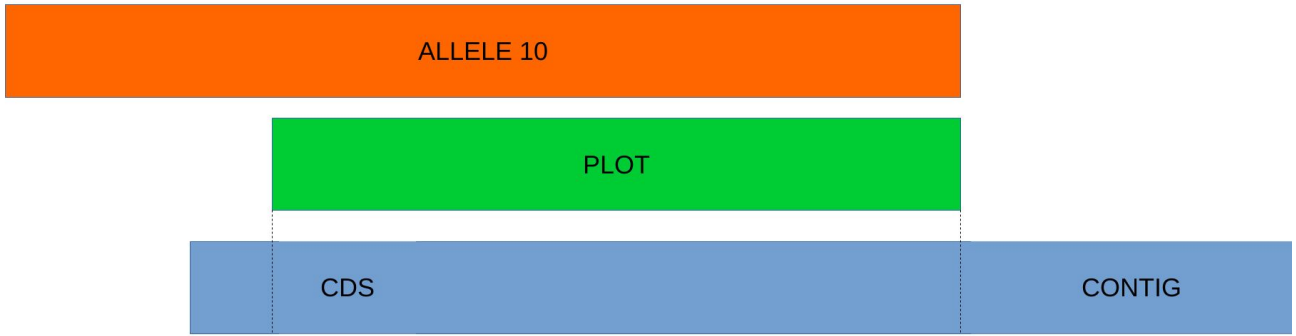



Query genome
Run prodigal on genome
CDSs
Exact Match
1
Exact match?
NIPHEM
>1
N
Translate CDS to AA
Genome BLASTP database
Schema (per locus)
Alleles (DNA)
Translate CDS to AA
Alleles (AA)
Self BLAST
BLASTP (per locus)
Calculate BSR
Add new allele (DNA) to Locus file
Allele definition
No blast match or BSR<=0.6
Y
LNF
N
>1 BSR>0.6 matches in different CDS
Y
NIPH
N
Size validation
Length 20% lesser than mode
ASM
Length 20% greater than mode
ALM
CDS position on tip of contig and possible larger matching allele
PLOT
Inferred New Allele

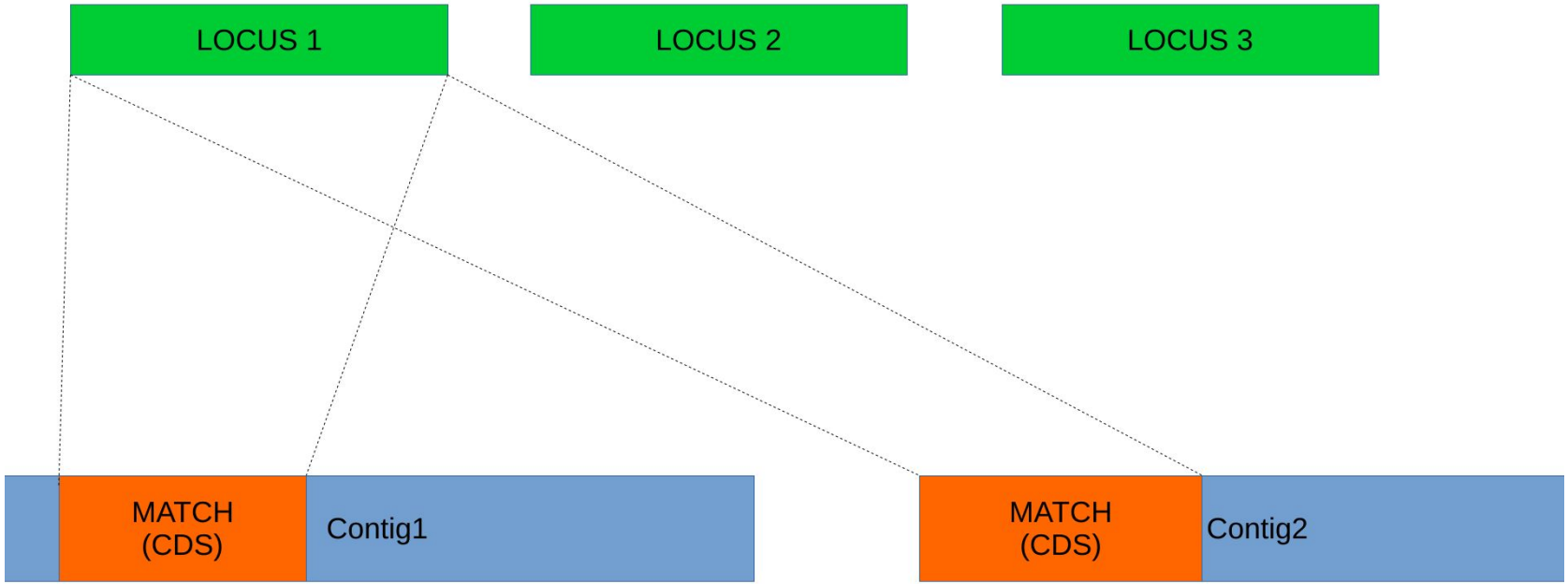
results_statistics.tsv

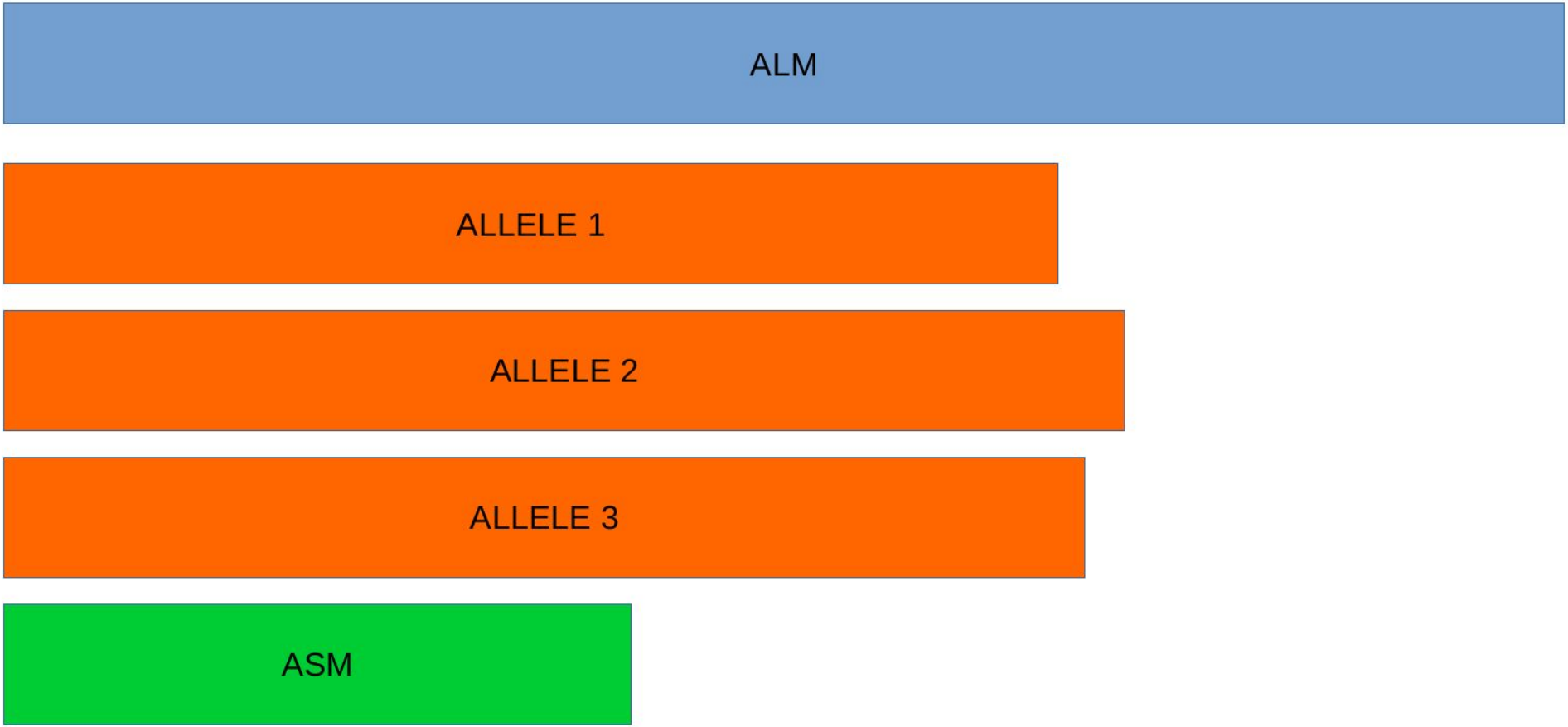
Genome	EXC	INF	LNF	PLOT	NIPH	ALM	ASM
NC_017162.fna	892	2319	1909	0	104	5	37
NC_011586.fna	1563	1697	1809	0	116	6	75

PLOT



NIPH/NIPHEM




ALM
ALLELE 1
ALLELE 2
ALLELE 3
ASM

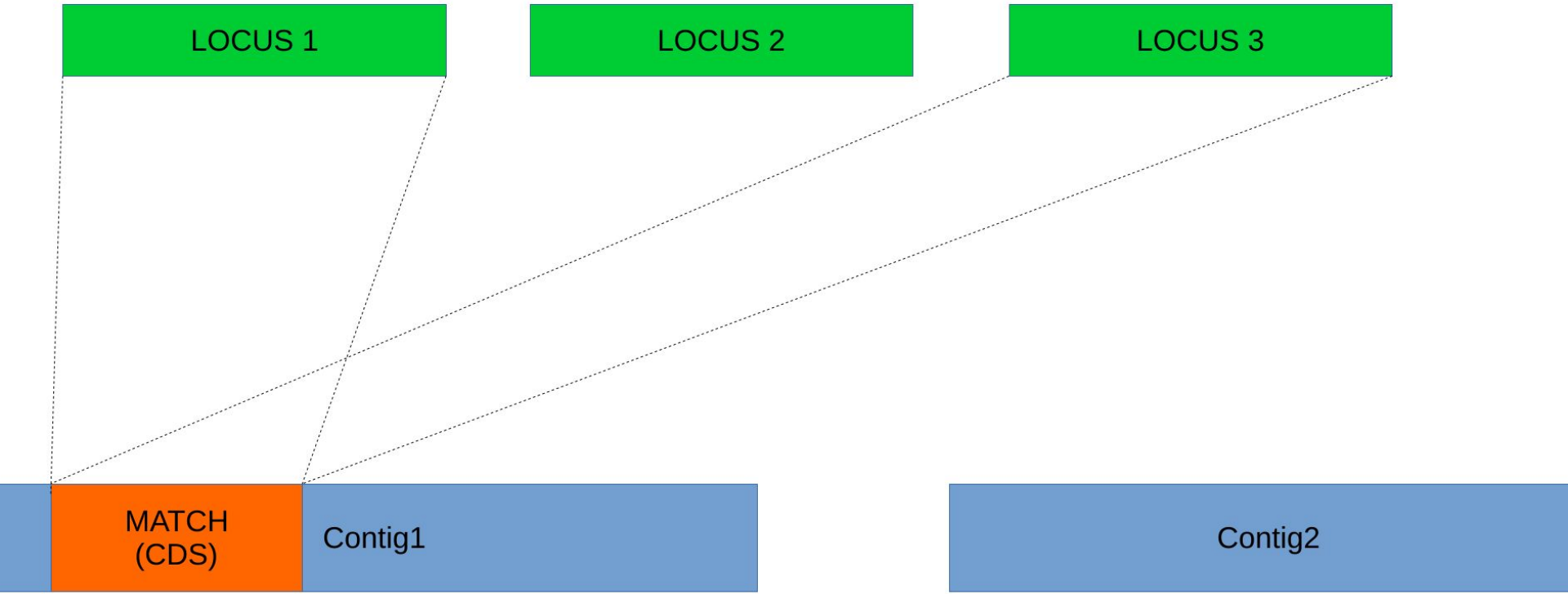
results_contigsInfo.tsv

FILE	GCA-000007265-protein3.fasta	GCA-000007265-protein25.fasta
905.1_ASM169490v1_genomic.fna	MBKW01000002.1&16193-14782&-	LNF
935.1_ASM169493v1_genomic.fna	MBKS01000045.1&108321-109730&+	ASM
945.1_ASM169494v1_genomic.fna	MBKR01000042.1&116556-117965&+	LNF

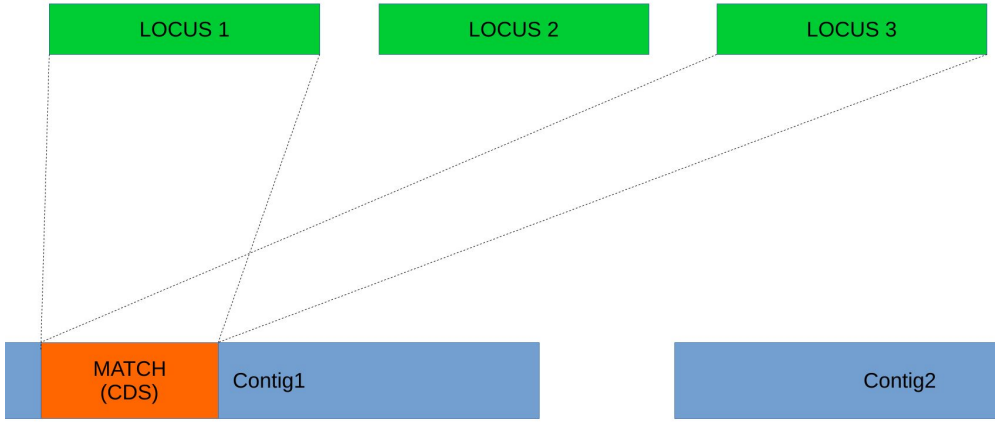
RepeatedLoci.tsv

gene	PC	NDC
GCA-000007265-protein46.fasta	1	0
GCA-000007265-protein128.fasta	1	0

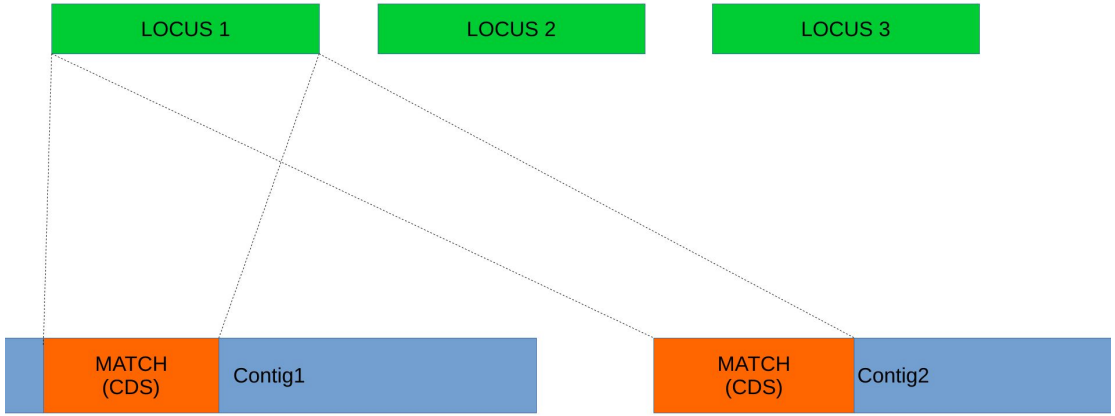
PARALOG DETECTION



PARALOG DETECTION



NIPH/NIPHEM



Allele Calling - other arguments

--ptf
Provide your own prodigal training file

--bsr 0.6
Minimum BSR for locus similarity (default: 0.6)

--fc
Force continue

--fr
Force reset

-v --verbose

cgMLST definition

cgMLST schemas are defined as the **set of loci** that are **present in all strains under analysis** or, due to sequencing/assembly limitations, **> 95% of strains** analyzed

cgMLST schema definition is based on pre-defined thresholds, only when a sufficient number of strains have been analyzed can the cgMLST schema be considered stable

The quality of draft genomes can impact profoundly the MLST schema definition. Therefore, chewBBACA offers a way to test the impact of each genome on the number of loci selected for inclusion in the final MLST schema, based on the number of missing loci when considering the original schema being tested.

Test genome allele call quality

	Locus 1	Locus 2	Locus 3	….
Genome 1	2	4	ALM	
Genome 2	1	PLOT	1	
Genome 3	LNF	1	2	

Test genome allele call quality

	Locus 1	Locus 2	Locus 3	….
Genome 1	2	4	ALM	
Genome 2	1	PLOT	1	
Genome 3	LNF	1	2	

X X X

Test genome allele call quality

	Locus 1	Locus 2	Locus 3	….
Genome 1	2	4	1	
Genome 2	1	3	1	
Genome 3	2	1	2	
Genome 4	PLOT	LNF	LNF	

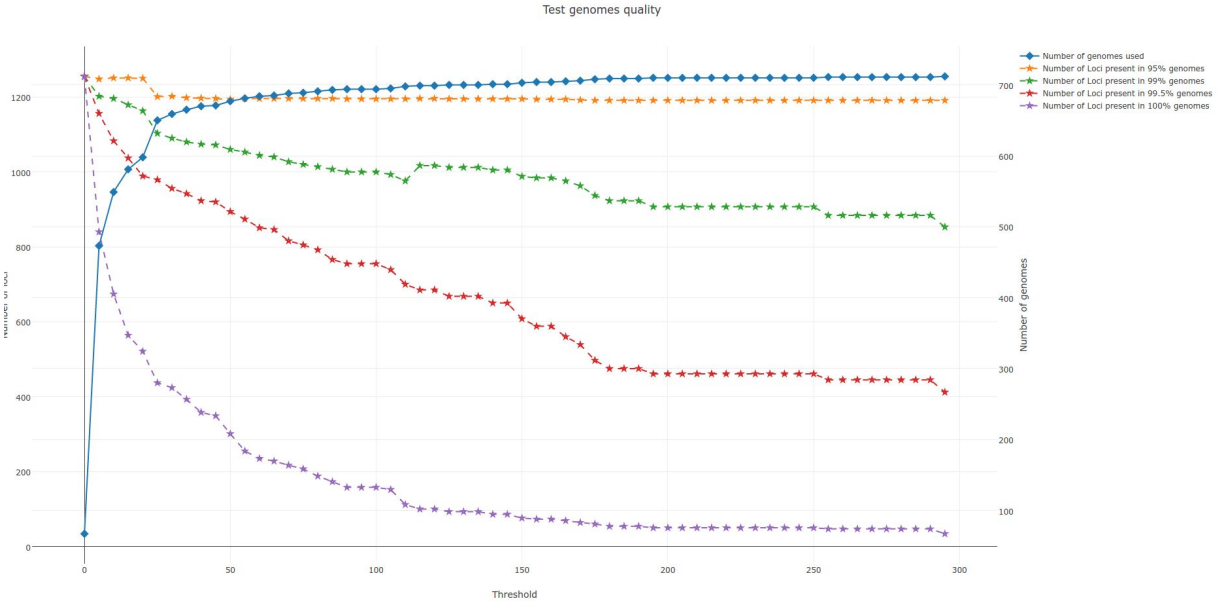
Test genome allele call quality

	Locus 1	Locus 2	Locus 3	….
Genome 1	2	4	1	
Genome 2	1	3	1	
Genome 3	2	1	2	
Genome 4	PLOT	LNF	LNF	

X X X

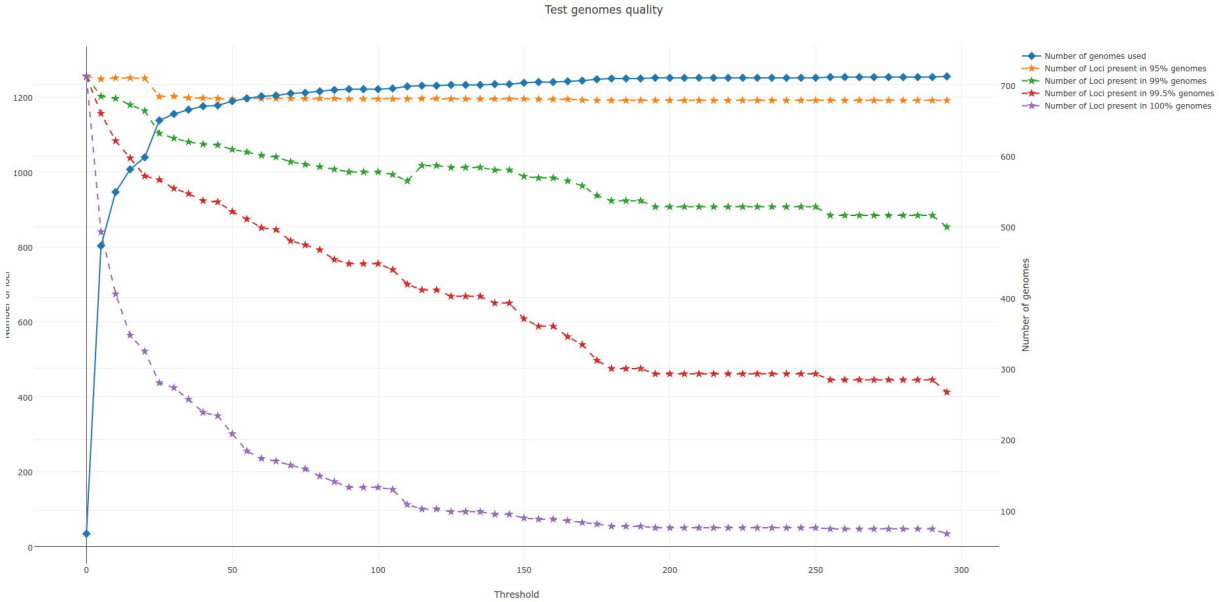
Test genome allele call quality

[example]



Task 4 - test genome allele call quality

chewBBACA.py TestGenomeQuality -i results/<results_20171207T150515>/results_alleles.tsv -o testcq -n 12 -s 5 -t 300



Task 5 - Extract the profile for phyloviz

chewBBACA.py ExtractCgMLST -i results/<results_20171207T150515>/results_alleles.tsv -o my_cgMLST -r results/results_20171207T150515/RepeatedLoci.txt -p 0.95

-p 0.95 Minimum percentage of genomes each included locus must be present in (e.g., set 0.95 to get a matrix with the loci that are present in at least 95% of the genomes). (default: 1)

-g genomes2remove.txt

-r loci2remove.txt

Output

cgMLST.tsv - profile output

cgMLSTschema.txt - list of loci that are considered for the cgMLST.tsv file

Presence_Abscence.tsv - presence/abscence from the input file

Schema eval

example

Task 6 - schema eval

chewBBACA.py SchemaEvaluator -o rms/Myschema.html --cpu 6

mickaelsilva@medicina.ulisboa.pt

https://github.com/B-UMMI/chewBBACA/wiki