

10iM-^iM-^T 06, 24 15:27

run-polap-function-select-contigs-by.sh

Page 1/7

```

1 #####
2 # This file is part of polap.
3 #
4 # polap is free software: you can redistribute it and/or modify it under the
5 # terms of the GNU General Public License as published by the Free Software
6 # Foundation, either version 3 of the License, or (at your option) any later
7 # version.
8 #
9 # polap is distributed in the hope that it will be useful, but WITHOUT ANY
10 # WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
11 # A PARTICULAR PURPOSE. See the GNU General Public License for more details.
12 #
13 # You should have received a copy of the GNU General Public License along with
14 # polap. If not, see <https://www.gnu.org/licenses/>.
15 #####
16 #####
17 # Ensure that the current script is sourced only once
18 source "$script_dir/run-polap-function-include.sh"
19 _POLAP_INCLUDE=$_polap_include "${BASH_SOURCE[0]}"
20 [[ -n "${_POLAP_INCLUDE}" ]] && return 0
21 declare "_POLAP_INCLUDE=1"
22 #
23 #####
24 #####
25 #####
26 # Selects contigs for an organelle-genome assembly.
27 #
28 # 1. We could select mitochondrial- or plastid-derived contigs using a contig annotation table.
29 # 2. We determine the range of sequencing depths for those candidate contigs: mean +/- sd \* 3.
30 # 3. For a given gfa of a genome assembly graph, subset the graph for selecting graph elements in the range.
31 # 4. Determine connected components in the subset.
32 # 5. Choose connected components with candidate edges.
33 #
34 # We need to read GFA files to manipulate.
35 # We need to determine connected components.
36 #####
37 function _run_polap_select-contigs-by() {
38     # Enable debugging if DEBUG is set
39     [ "$DEBUG" -eq 1 ] && set -x
40     _polap_log_function "Function start: $(echo $FUNCNAME | sed s/_run_polap_/)"
41
42     # Set verbosity level: stderr if verbose >= 2, otherwise discard output
43     local _polap_output_dest="/dev/null"
44     [ "${_arg_verbose}" -ge "${_polap_var_function_verbose}" ] && _polap_output_dest="/dev/stderr"
45
46     # Grouped file path declarations
47     source "$script_dir/polap-variables-mtcontig.sh" # '.' means 'source'
48
49     # Print help message if requested
50     help_message=$(
51
```

10iM-^iM-^T 06, 24 15:27

run-polap-function-select-contigs-by.sh

Page 2/7

```

52         cat <<HEREDOC
53 # Selects contigs using three features using total lengths of contigs.
54 #
55 # Use the total length in the cumulative distribution of contig lengths
56 # to select the lower and upper bounds of contig lengths.
57 #
58 # To identify seed contigs of mitochondrial origin,
59 # a whole-genome assembly is evaluated for three criteria:
60 # 1) the presence of mitochondrial or plastid genes,
61 # 2) the number of read coverage, and
62 # 3) the connectivity of contigs in the genome assembly graph.
63 #
64 # 1. We could select mitochondrial- or plastid-derived contigs using a contig annotation table.
65 # 2. We determine the range of sequencing depths for those candidate contigs: mean +/- sd \* 3.
66 #     2.1 Construct the cumulative distribution of contig lengths.
67 #     2.2 Given L1=3Mb, determine the lower bound of the contig length.
68 #     2.3 Given L2=300 kb, determine the upper bound of the contig length.
69 # 3. For a given gfa of a genome assembly graph, subset the graph for selecting graph elements in the range.
70 # 4. Determine connected components in the subset.
71 # 5. Choose connected components with candidate edges.
72 #
73 # Arguments:
74 #   -i $INUM: source Flye (usually whole-genome) assembly number
75 #   -j $JNUM: destination Flye organelle assembly number
76 #   --select-contig: 1 ~ 5
77 #
78 # Inputs:
79 #   ${_polap_var_assembly_graph_final_gfa}
80 #   ${_polap_var_annotation_table}
81 #
82 # Outputs:
83 #   $MTCONTIGNAME
84 #   "${_polap_var_mtcontig_annotated}"
85 #
86 # See:
87 #   run-polap-select-contigs-by-table-1.R for the description of --select-contig option
88 Example: $(basename $0) ${_arg_menu[0]} [-i|--inum <arg>] [-j|--jnum <arg>] [--select-contig <number>]
89 Example: $(basename $0) ${_arg_menu[0]} -o PRJNA914763 -i 0 -j 5 --select-contig 5
90 HEREDOC
91 )
92
93 # Display help message
94 [[ ${_arg_menu[1]} == "help" ]] && _polap_echo0 "${help_message}" && exit $EXIT_SUCCESS
95
96 # Display the content of output files
97 if [[ "${_arg_menu[1]}" == "view" ]]; then
98
99     case "${_arg_select_contig}" in
100     1 | 3)
101         check_file_existence "${_polap_var_mtcontig_annotated}"
102         _polap_log0_cat "${_polap_var_mtcontig_annotated}"

```

10iM-^iM-^T 06, 24 15:27

run-polap-function-select-contigs-by.sh

Page 3/7

```

103         check_file_existence "${_polap_var_mtcontig_stats}"
104         _polap_log0_cat "${_polap_var_mtcontig_stats}"
105         ;;
106     2 | 4 | 5)
107         check_file_existence "${_polap_var_mtcontig_mixfit}"
108         _polap_log0_cat "${_polap_var_mtcontig_mixfit}"
109         check_file_existence "${_polap_var_mtcontig_annotated}"
110         _polap_log0_cat "${_polap_var_mtcontig_annotated}"
111         check_file_existence "${_polap_var_mtcontig_stats}"
112         _polap_log0_cat "${_polap_var_mtcontig_stats}"
113         check_file_existence "${_polap_var_mtcontig_mixfit}"
114         _polap_log0_cat "${_polap_var_mtcontig_mixfit}"
115         ;;
116     *)
117         echo "Invalid input!"
118         ;;
119     esac
120
121     _polap_log2 "Function end: $(echo $FUNCNAME | sed s/_run_polap_/)"
122     # Disable debugging if previously enabled
123     [ "$DEBUG" -eq 1 ] && set +x
124     exit $EXIT_SUCCESS
125 fi
126
127 _polap_log0 "selecting seed contigs using $(echo $FUNCNAME | sed s/_run_polap_/) ${INUM} -> ${JNUM} with type ${_arg_select_contig}"
128
129 # Check for required files
130 check_file_existence "${_polap_var_assembly_graph_final_gfa}"
131 check_file_existence "${_polap_var_annotation_table}"
132 _polap_log0 " input1: ${_polap_var_assembly_graph_final_gfa}"
133 _polap_log0 " input2: ${_polap_var_annotation_table}"
134
135 # Clean and create working directory
136 _polap_log1 " delete and create dir:${_polap_var_mtcontigs}"
137 _polap_log3_cmd rm -rf "${_polap_var_mtcontigs}"
138 _polap_log3_cmd mkdir -p "${_polap_var_mtcontigs}"
139
140 # Step 1: Determine the depth range using the cumulative length distribution.
141 # Step 1: Select contigs based on genes
142 _polap_log1 " select-contig type: ${_arg_select_contig}"
143 _polap_log1 "run-polap-select-contigs-by-table-1.R"
144 _polap_log2_file " input1: ${_polap_var_annotation_table}"
145 _polap_log2_file " output-base1: ${_polap_var_mtcontig_base}"
146 case "${_arg_select_contig}" in
147 1 | 3)
148     "$WDIR"/run-polap-select-contigs-by-table-1.R \
149         -t "${_polap_var_annotation_table}" \
150         -o "${_polap_var_mtcontig_base}" \
151         -c -d 10 \
152         2>"${_polap_output_dest}"
153     ;;

```

10ìM-^M-^T 06, 24 15:27

run-polap-function-select-contigs-by.sh

Page 4/7

```

154         2 | 4)
155             "$WDIR"/run-polap-select-contigs-by-table-1.R \
156                 -t "${_polap_var_annotation_table}" \
157                 -o "${_polap_var_mtcontig_base}" \
158                 -c -d 10 \
159                 -r \
160                 2>"$_polap_output_dest"
161             ;;
162     5)
163         "$WDIR"/run-polap-select-contigs-by-table-1.R \
164             -t "${_polap_var_annotation_table}" \
165             -o "${_polap_var_mtcontig_base}" \
166             -c -d 10 \
167             -r \
168             -s \
169             2>"$_polap_output_dest"
170         ;;
171     *)
172         echo "Invalid input!"
173         ;;
174 esac
175
176 _polap_log2_file " output1: ${_polap_var_mtcontig_stats}"
177 _polap_log2_file " output2: ${_polap_var_mtcontig_annotated}"
178
179 case "${_arg_select_contig}" in
180     1 | 2)
181         # Save the first column (contig names) to the output file
182         if [ -s "${_polap_var_mtcontig_annotated}" ]; then
183             cut -f1 "${_polap_var_mtcontig_annotated}" |
184                 sort | uniq >"${MTCONTIGNAME}"
185             _polap_log1_file "output: ${MTCONTIGNAME}"
186         else
187             >"${MTCONTIGNAME}"
188             _polap_log1_file "output: empty ${MTCONTIGNAME}"
189         fi
190         _polap_log2 "Function end: $(echo $FUNCNAME | sed s/_run_polap_/)"
191         [ "$DEBUG" -eq 1 ] && set +x
192         return
193         ;;
194     3 | 4 | 5)
195         if [ ! -s "${_polap_var_mtcontig_annotated}" ]; then
196             >"${MTCONTIGNAME}"
197             _polap_log1_file "output: empty ${MTCONTIGNAME}"
198             _polap_log2 "Function end: $(echo $FUNCNAME | sed s/_run_polap_/)"
199             [ "$DEBUG" -eq 1 ] && set +x
200             return
201         fi
202         ;;
203     *)
204         echo "Invalid input!"

```

10ìM-^M-^T 06, 24 15:27

run-polap-function-select-contigs-by.sh

Page 5/7

```

205         ;;
206     esac
207
208     # Handle case with single starting contig
209     local mtcontig_count=$(wc -l <"${_polap_var_mtcontig_annotated}")
210     if [ "${mtcontig_count}" -eq 1 ]; then
211         cut -f1 "${_polap_var_mtcontig_annotated}" >"${MTCONTIGNAME}"
212         _polap_log2_log "single starting contig"
213         _polap_log1_file "output: ${MTCONTIGNAME}"
214         _polap_log2 "Function end: $(echo $FUNCNAME | sed s/_run_polap_/)"
215         [ "$DEBUG" -eq 1 ] && set +x
216         return
217     fi
218
219     # Extract sequences and filter GFA data
220     _polap_log2 "creating GFA without sequence data"
221     gfatools view -S "${_polap_var_assembly_graph_final_gfa}" \
222         >"${_polap_var_gfa_all}" \
223         2>"${_polap_output_dest}"
224     _polap_log2_file "${_polap_var_gfa_all}"
225
226     _polap_log2 "extracting sequence part of GFA"
227     gfatools view -S "${_polap_var_assembly_graph_final_gfa}" \
228         2>"${_polap_output_dest}" |
229         grep "^S" >"${_polap_var_gfa_seq_part}"
230     _polap_log2_file "${_polap_var_gfa_seq_part}"
231
232     # Filter edges in GFA using depths.
233     _polap_log2 "filtering GFA sequence part using depth range"
234     "${WDIR}"/run-polap-select-contigs-by-depth-length-2-gfa-filter.R \
235         "${_polap_var_gfa_seq_part}" \
236         "${_polap_var_mtcontig_stats}" \
237         "${_polap_var_gfa_seq_filtered}" \
238         "${_polap_var_gfa_seq_filtered_range}" \
239         2>"${_polap_output_dest}"
240
241     _polap_log2_file "${_polap_var_gfa_seq_filtered}"
242
243     # Recreate GFA based on filtered edge sequences.
244     _polap_log2 "subsetting GFA using the depth-filtered GFA sequence part"
245     cut -f1 "${_polap_var_gfa_seq_filtered}" >"${_polap_var_gfa_seq_filtered_edge}"
246     gfatools view -S \
247         -l @"${_polap_var_gfa_seq_filtered_edge}" \
248         "${_polap_var_assembly_graph_final_gfa}" 2>/dev/null \
249         >"${_polap_var_gfa_filtered}"
250
251     _polap_log2_file "${_polap_var_gfa_filtered}"
252
253     # Prepare links for finding connected components.
254     grep "^L" "${_polap_var_gfa_filtered}" | cut -f2,4 >"${_polap_var_gfa_links}"
255     _polap_log2_file "${_polap_var_gfa_links}"

```

10ìM-^M-^T 06, 24 15:27

run-polap-function-select-contigs-by.sh

Page 6/7

```

256
257 # Run R script to analyze GFA links
258 _polap_log2 "preparing for finding connected components"
259 "$WDIR"/run-polap-select-contigs-3-gfa-links.R \
260     "${_polap_var_mtcontig_annotated}" \
261     "${_polap_var_gfa_links}" \
262     "${_polap_var_gfa_links_number}" \
263     "${_polap_var_gfa_links_order}" \
264     "${_polap_var_gfa_links_contig}" \
265     "${_polap_var_gfa_links_contig_na}" \
266     2>"$_polap_output_dest"
267
268 _polap_log2_file "${_polap_var_gfa_links_number}"
269 _polap_log2_file "${_polap_var_gfa_links_order}"
270 _polap_log2_file "${_polap_var_gfa_links_contig}"
271 _polap_log2_file "${_polap_var_gfa_links_contig_na}"
272
273 # Find connected components using Python script
274 _polap_log2 "finding connected components by the depth-filtered contigs"
275 python "$WDIR"/run-polap-select-contigs-4-find-connected-components.py \
276     "${_polap_var_gfa_links_number}" \
277     "${_polap_var_gfa_links_contig}" \
278     "${_polap_var_gfa_links_seed}" \
279     2>"$_polap_output_dest"
280
281 _polap_log2_file "${_polap_var_gfa_links_seed}"
282
283 # Choose final mitochondrial contigs
284 _polap_log2 "converting the depth-filtered contigs in edge with numbers"
285 "$WDIR"/run-polap-select-contigs-5-gfa-mtcontig.R \
286     "${_polap_var_gfa_links_seed}" \
287     "${_polap_var_gfa_links_order}" \
288     "${_polap_var_gfa_links_mtcontig}" \
289     2>"$_polap_output_dest"
290
291 _polap_log2_file "${_polap_var_gfa_links_mtcontig}"
292
293 _polap_log2 "concatenating the depth-filtered edges and NA edges?"
294 cat "${_polap_var_gfa_links_mtcontig}" "${_polap_var_gfa_links_contig_na}" |
295     sort | uniq >"${MTCONTIGNAME}"
296 _polap_log1_file "output: ${MTCONTIGNAME}"
297
298 "$WDIR"/run-polap-select-contigs-by-table-2.R \
299     -t "${_polap_var_annotation_table}" \
300     -m "${MTCONTIGNAME}" \
301     -o "${_polap_var_mtcontig_base}" \
302     2>"$_polap_output_dest"
303
304 # _polap_log2_file " output1: ${_polap_var_mtcontig_stats}"
305 _polap_log2_file " output2: ${_polap_var_mtcontig_annotated}"
306

```

10ìM-^[M-^T 06, 24 15:27

run-polap-function-select-contigs-by.sh

Page 7/7

```
307     _polap_log2 "Function end: $(echo $FUNCNAME | sed s/_run_polap_/)"
308     # Disable debugging if previously enabled
309     [ "$DEBUG" -eq 1 ] && set +x
310 }
```