PSODA

Open Source Phylogenetic Search and DNA Analysis

Quinn Snell, Mark Clement, Kenneth Sundberg

Brigham Young University

{snell,clement,kasundberg}@cs.byu.edu

http://dna.cs.byu.edu/psoda --- svn co http://dna.cs.byu.edu/opensvn/psoda

Abstract

PSODA (sō-də) is an open source (GPL v2) sequence analysis package that implements sequence alignment using biochemical properties, phylogeny search with parsimony or maximum likelihood criteria and selection detection using biochemical properties (TreeSAAP). PSODA is compatible with PAUP* and the search algorithms are competitive with those in PAUP*. PSODA also adds a basic scripting language to the PAUP block, making it possible to easily create advanced meta-searches. Because PSODA is open-source, we have also been able to easily add in advanced search techniques and characterize the benefits of various optimizations. PSODA is available for Macintosh OS X, Windows, and Linux.
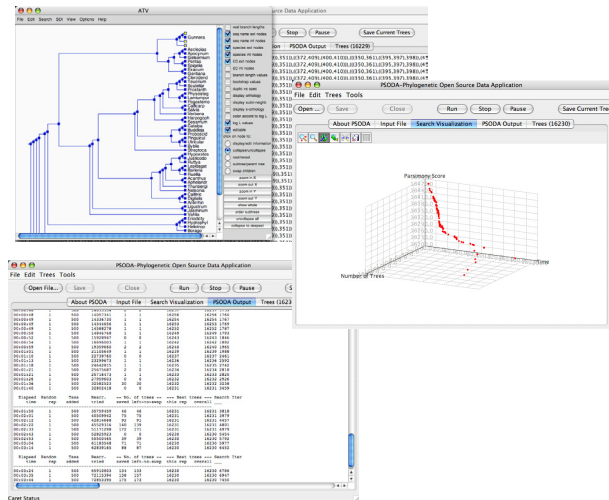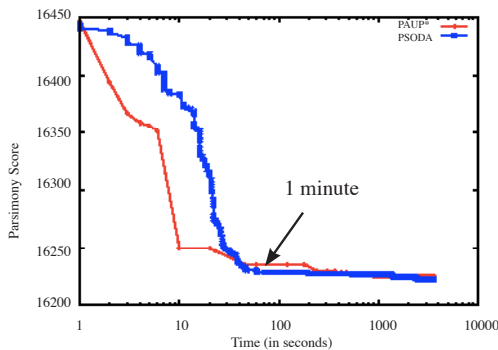
Advantages of PSODA

- High Performance
- Open Source (-: FREE :-)
- Modular Design (easy algorithm development)
- Advanced Scripting Language
  - makes advanced meta-searches simple
- Reads and Executes PAUP nexus files
- PSODA is competive with PAUP*

PSODA Features

- Parsimony and Likelihood (RAxML) search
- Baysian methods (Mr. Bayes)
- Consensus (strict and majority rules)
- Selection Detection (TreeSAAP)
- Graphical User Interface
- Binaries for Mac OS X, Windows and Linux
- Object-oriented C++
  - easy to contribute to new algorithm development
- Available via subversion
  - svn checkout http://dna.cs.byu.edu/opensvn/psoda

Advanced Scripting Language

- Added functionality for PAUP blocks.
- Decision Statements & Loops
- Advanced Functions & User-defined Functions
- Easily Extensible
- Easy scripting of advanced meta-searches such as:
  - Ratchet (Parsimony and Likelihood)
  - DCM and more.

PAUP* Ratchet

```
BEGIN PAUP;
set criterion=parsimony ;
set maxtrees=1
increase = no;
hsearch start=stepwise swap=TBR ;
weights 2 : 7 14 17 26 27 31 34 45 50 52 54 57 60 63 64 65 73 77 86 91 92 100 103 107 115 117 121
122 124 127 131 133 134 140 142 155 156 163 173 176 183 185 187 195 197 198 202 204 209 219 221 222
225 226 230 237 238 240 241 244 248 252 254 255 258 269 276 279 283 284 291 294 295 297 309 310 315
319 321 323 325 324 342 346 349 350 351 353 354 355 359 360 363 366 370 377 378 390 393 394 398 408
412 413 418 419 423 425 426 428 429 432 435 456 467 475 479 482 484 489 492 493 497 498 500;
hsearch start=current swap=TBR
;
set maxtrees=1;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 1 17 20 21 29 33 35 39 42 50 57 59 60 71 73 80 86 93 96 97 99 100 102 111 121 125 126
141 142 148 149 155 157 163 168 174 179 180 184 191 203 208 212 216 219 220 225 226 231 242
243 251 253 256 275 267 260 265 269 272 274 277 282 286 287 288 291 293 294 307 305 306 303 315 314
318 324 326 327 329 334 340 346 351 352 356 360 364 370 372 373 379 380 384 390 384 398 401 408 412
414 421 423 426 428 429 441 442 444 448 455 459 460 462 472 473 475 477 478 480 484 488 493;
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 2 9 11 14 16 18 20 24 28 36 40 43 45 47 59 60 62 64 65 70 72 84 90 92 105 107 111 112
118 122 129 130 135 138 144 148 152 153 156 161 162 167 172 173 177 190 197 200 207 209 213 215 217
218 224 234 245 249 250 251 253 254 255 257 264 275 274 281 284 286 293 303 305 315 318 322 324 329
336 347 349 351 352 355 356 358 360 365 367 368 371 374 377 378 381 382 407 409 410 411 412 413 418
421 424 425 426 437 442 444 447 448 452 456 461 462 464 474 475 477 478 480 484 488 491;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 1 5 17 22 26 31 34 39 44 46 65 71 73 79 80 85 92 93 95 98 100 101 107 114 119 127 131
137 138 139 142 145 146 150 151 152 155 161 162 167 170 175 176 177 188 191 197 198 211 212 217 220
226 232 236 238 239 240 242 245 252 257 268 269 270 275 281 283 287 300 301 305 306 309 311 318
323 324 326 333 336 335 342 345 349 350 351 361 363 367 369 371 372 374 384 388 395 404 408 410 425
426 427 430 431 432 434 435 439 443 445 447 452 455 457 461 465 467 470 474 483 485 490;
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 1 2 3 9 20 22 24 25 26 28 30 32 34 38 43 46 49 58 60 66 68 70 72 75 87 88 91 97 99 103 105
109 112 115 118 123 139 140 144 146 152 160 162 165 168 180 184 194 197 200 204 207 214 217 220 222
223 225 233 240 244 249 250 251 257 260 262 270 275 280 286 290 291 292 297 298 301 307 310 313 314
328 331 332 324 334 336 338 339 342 347 355 360 373 378 380 381 389 395 396 397 401 403 404 407
410 413 425 430 447 454 455 460 461 464 472 478 482 483 487 488 489 492 494 498;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 1 2 3 6 12 25 29 30 37 50 56 60 61 63 64 67 68 69 70 75 79 85 87 95 98 102 103 108 110 120
123 125 127 129 131 142 145 157 164 168 170 172 175 176 180 191 194 195 196 197 201 202 204
205 206 211 214 216 227 230 235 236 240 243 244 250 252 255 257 263 266 272 281 289 304 307 308 309
310 327 328 329 336 338 342 350 353 354 356 357 358 368 380 382 387 389 397 400 405 407 408 410
413 418 421 422 423 424 440 444 446 451 452 455 457 463 468 470 476 480 485 495;
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 1 5 6 19 24 27 31 32 33 34 35 39 40 43 49 58 59 61 63 70 73 89 90 92 94 97 98 113 116 118 121
126 127 131 140 141 144 148 166 165 169 170 173 175 176 177 179 181 183 185 191 194 202 203 204 205
207 211 213 215 218 220 233 234 238 245 252 254 259 260 265 269 272 278 280 284 287 288 292 295 297
302 304 306 310 313 315 317 319 328 331 334 335 336 342 346 347 351 357 362 365 377 384 388 389 390
403 411 419 426 438 440 443 444 447 455 460 463 464 467 486 491 493 494 499 500;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
```

Repeated text must continue. However, it is unclear when to stop.

PSODA Likelihood Ratchet

```
BEGIN PAUP;

begin randomReweight
  numChars = getWeightsLength();
  numWeights = numChars / percent;

  j = 0;
  while (j < numweights)
    weight = random(max = range);
    col = random(max = numChars) + 1;
    weights weight:col;
    j++;
  endwhile;
end;

set maxtrees = 1 nreps = 5;
set criterion=parsimony;
hsearch start = stepwise swap = tbr;
range = 3;

while (true)
  randomReweight(percent = 10);
  set criterion=parsimony;
  hsearch start = current swap = tbr;

  weights reset;

  set criterion=likelihood;
  hsearch start = current swap = tbr;
endwhile;

end;
```

http://dna.cs.byu.edu/psoda