R is a high-level programming language for statistical computation. It is widely used in bioinformatics:
- mixture analysis
- gene shaving
- clinical trial analysis
- microarray analysis

R is convenient but slow. Researchers shouldn’t have to translate scripts by hand into Fortran or C to achieve acceptable performance.

RCC compiles R into C with some performance gains already. New analysis and optimization techniques

### R’s need for speed: vector operations inefficient

Excerpt from Bayesian mixture analysis code (M.D. Anderson Cancer Center):

The variables ss and r are arrays of 800 elements each.

```r
R> ss[r==1] <- ss[r==1] - 1
```

The R interpreter allocates three 800-element temporary vectors to interpret this line of code. All three allocations are unnecessary.

### R’s unique features

- Scheme-like scoping
- Imperative call-by-need semantics

### High-level vector operations

- `t1`: vector of TRUE and FALSE values indicating elements of `r` equal to 1
- `t2`: subset of `ss` corresponding to `t1`
- `t3`: vector difference: subtract 1 from each element of `t2`

### Future optimization (example)

Type inference eliminates redundant checks

Excerpt from R interpreter code for arithmetic operations:

```r
else if (TYPEOF(x) == REALSXP || TYPEOF(y) == REALSXP) {
  COERCE_IF_NEEDED(x, REALSXP, xpi);
  COERCE_IF_NEEDED(y, REALSXP, ypi);
  else val = real_binary(PRIMVAL(op), x, y);
  return val;
  break;
}
```

### Acknowledgments

Thanks to Cheryl McCosh, Walid Taha, Keith Cooper, Kevin Coombes, and Keith Baggerly for many helpful discussions.

Funding for this research comes from the National Partnership for Advanced Computational Infrastructure (NPACI).