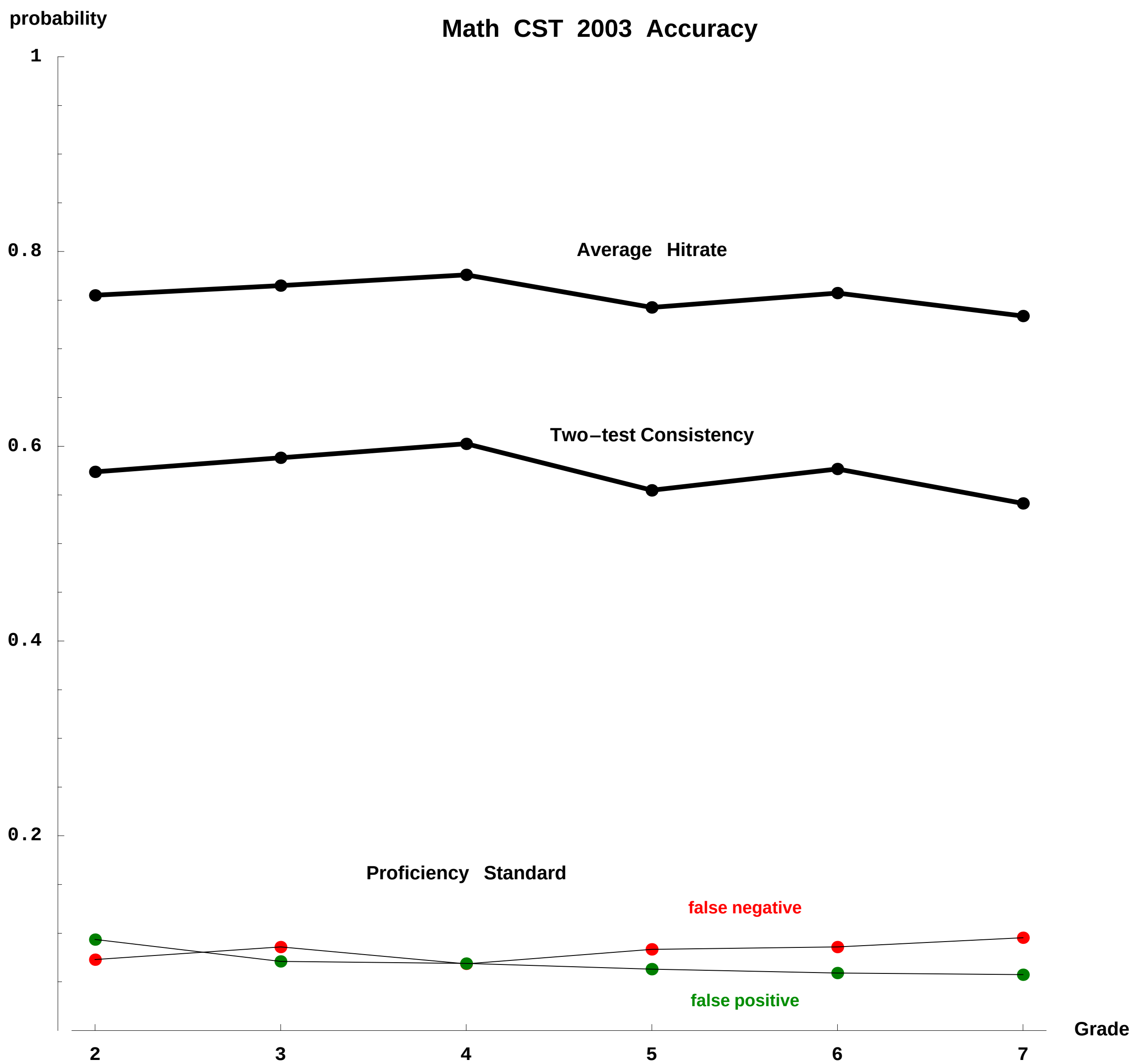
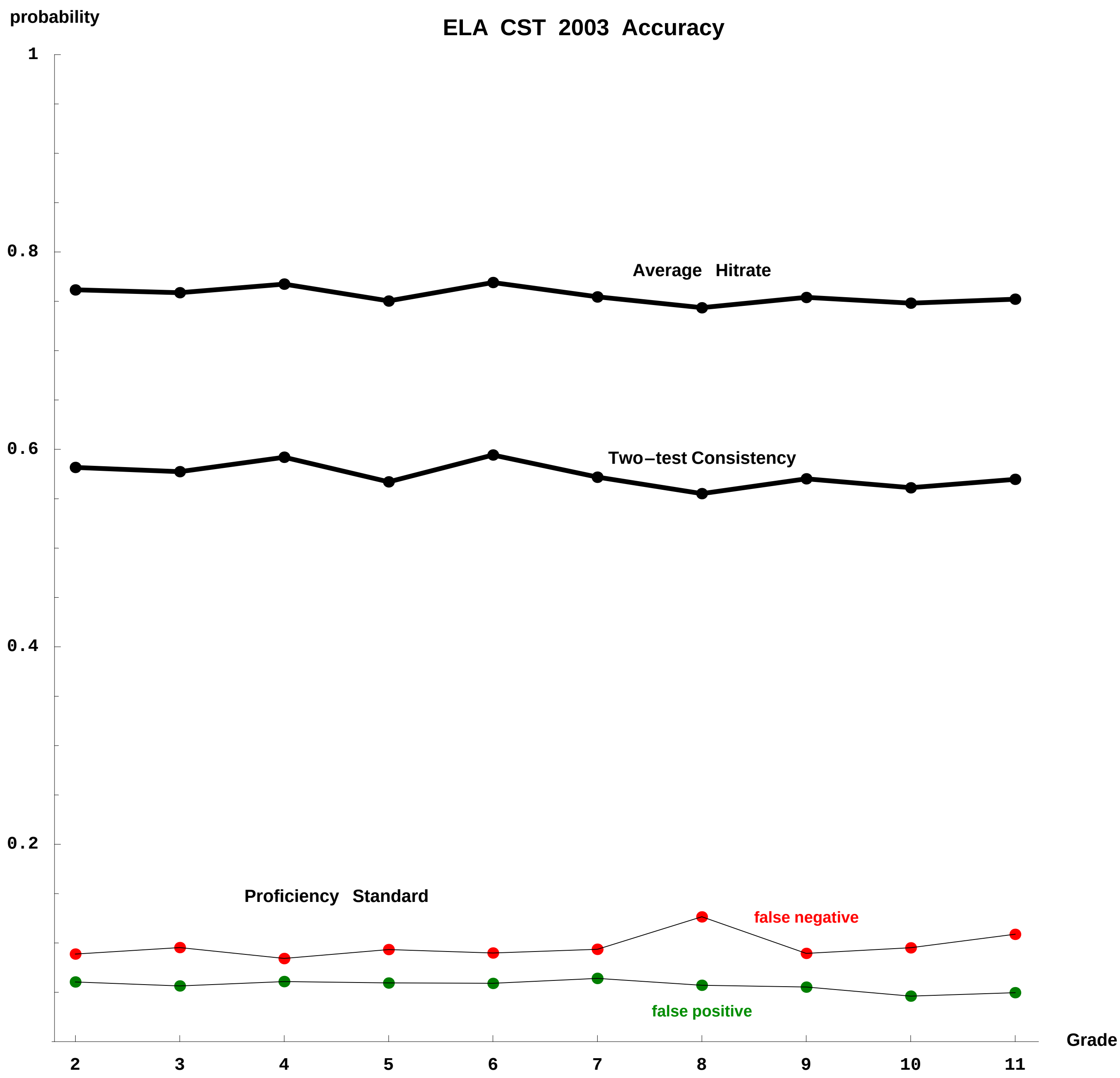
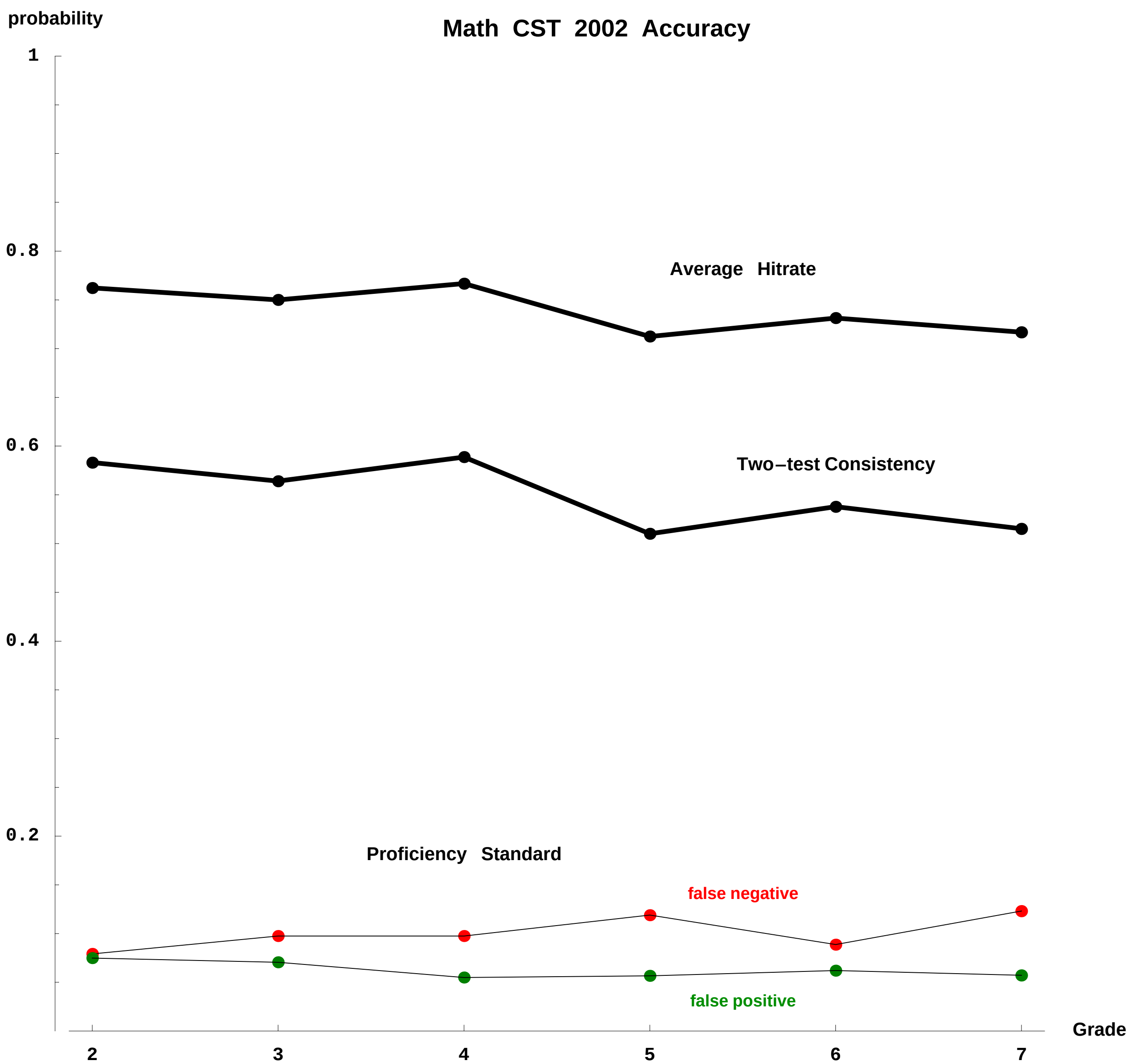
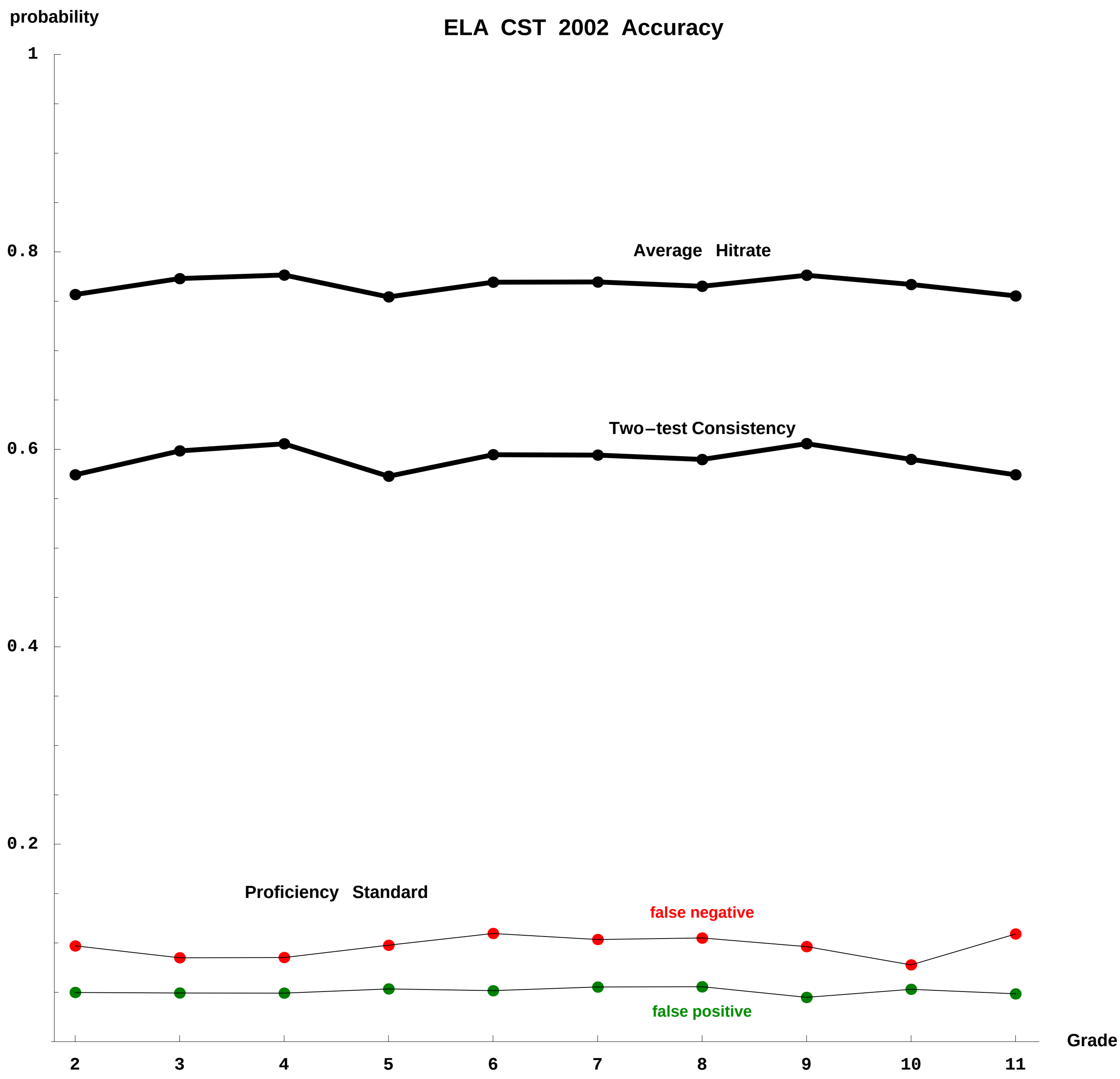

probability
Math CST 2003 Accuracy
1
0.8
0.6
0.4
0.2
Average Hitrate
Two−test Consistency
Proficiency Standard
false negative
false positive
Grade
2
3
4
5
6
7


ELA CST 2003 Accuracy
probability
1
0.8
0.6
0.4
0.2
Average Hitrate
Two–test Consistency
Proficiency Standard
false negative
false positive
Grade
2
3
4
5
6
7
8
9
10
11



probability
Math CST 2002 Accuracy
1
0.8
0.6
0.4
0.2
Average Hitrate
Two−test Consistency
Proficiency Standard
false negative
false positive
Grade
2
3
4
5
6
7



probability
ELA CST 2002 Accuracy
1
0.8
0.6
0.4
0.2
Average Hitrate
Two−test Consistency
Proficiency Standard
false negative
false positive
Grade
2
3
4
5
6
7
8
9
10
11

**Table 1.**
**Math CST 2003 Hit-rate Accuracy: overall and by performance level**

| Grade | Average | FBB | BB | B | P | A |
|---|---|---|---|---|---|---|
| 2 | 0.7551 | 0.8011 | 0.7833 | 0.6723 | 0.7291 | 0.8383 |
| 3 | 0.765 | 0.8109 | 0.7905 | 0.7073 | 0.7321 | 0.8534 |
| 4 | 0.776 | 0.7932 | 0.7919 | 0.7559 | 0.763 | 0.8012 |
| 5 | 0.7426 | 0.7136 | 0.7024 | 0.7012 | 0.7996 | 0.8655 |
| 6 | 0.7573 | 0.5709 | 0.7322 | 0.7455 | 0.7913 | 0.873 |
| 7 | 0.7337 | 0.6698 | 0.6728 | 0.736 | 0.8022 | 0.8291 |

**Table 2.**
**ELA CST 2003 Hit-rate Accuracy: overall and by performance level**

| Grade | Average | FBB | BB | B | P | A |
|---|---|---|---|---|---|---|
| 2 | 0.7617 | 0.8156 | 0.6901 | 0.7787 | 0.7523 | 0.8052 |
| 3 | 0.7588 | 0.8454 | 0.7079 | 0.7655 | 0.743 | 0.7554 |
| 4 | 0.7676 | 0.6929 | 0.6896 | 0.8024 | 0.7439 | 0.8448 |
| 5 | 0.7504 | 0.6929 | 0.6308 | 0.7924 | 0.7753 | 0.8058 |
| 6 | 0.7691 | 0.8033 | 0.661 | 0.7995 | 0.7518 | 0.8281 |
| 7 | 0.7546 | 0.7913 | 0.656 | 0.7769 | 0.7744 | 0.7843 |
| 8 | 0.7436 | 0.8077 | 0.6708 | 0.7727 | 0.7195 | 0.7843 |
| 9 | 0.7541 | 0.779 | 0.6885 | 0.7773 | 0.7385 | 0.8067 |
| 10 | 0.7482 | 0.7388 | 0.7001 | 0.8002 | 0.7356 | 0.7431 |
| 11 | 0.7523 | 0.8527 | 0.6784 | 0.7793 | 0.6981 | 0.7597 |

**Table 3.**
**Math CST 2002 Hit-rate Accuracy: overall and by performance level**

| Grade | Average | FBB | BB | B | P | A |
|---|---|---|---|---|---|---|
| 2 | 0.7621 | 0.8203 | 0.7712 | 0.7058 | 0.7477 | 0.8479 |
| 3 | 0.75 | 0.8138 | 0.7736 | 0.6981 | 0.7427 | 0.7923 |
| 4 | 0.7667 | 0.7278 | 0.8096 | 0.7649 | 0.7288 | 0.7776 |
| 5 | 0.7124 | 0.5949 | 0.6871 | 0.7036 | 0.7607 | 0.8341 |
| 6 | 0.7314 | 0.5912 | 0.7049 | 0.7245 | 0.7634 | 0.8504 |
| 7 | 0.7167 | 0.6666 | 0.6818 | 0.7139 | 0.7695 | 0.7872 |

**Table 4.**
**ELA CST 2002 Hit-rate Accuracy: overall and by performance level**

| Grade | Average | FBB | BB | B | P | A |
|---|---|---|---|---|---|---|
| 2 | 0.7568 | 0.8088 | 0.7062 | 0.7875 | 0.7434 | 0.7236 |
| 3 | 0.773 | 0.831 | 0.7315 | 0.7762 | 0.7613 | 0.797 |
| 4 | 0.7765 | 0.7507 | 0.6895 | 0.8225 | 0.7638 | 0.8287 |
| 5 | 0.7545 | 0.7282 | 0.6535 | 0.8025 | 0.7639 | 0.8037 |
| 6 | 0.7694 | 0.8444 | 0.6828 | 0.799 | 0.7506 | 0.7663 |
| 7 | 0.7695 | 0.8492 | 0.7071 | 0.7769 | 0.7713 | 0.7423 |
| 8 | 0.7653 | 0.8416 | 0.6697 | 0.8059 | 0.7146 | 0.8168 |
| 9 | 0.7764 | 0.8437 | 0.6931 | 0.8025 | 0.7371 | 0.82 |
| 10 | 0.767 | 0.8069 | 0.7114 | 0.7883 | 0.7353 | 0.8171 |
| 11 | 0.7553 | 0.8472 | 0.6804 | 0.7833 | 0.6973 | 0.7903 |

**Table 5.**
**Math CST 2003 Test-retest Accuracy: overall and by performance level**

| Grade | Average Correct Matches | Average Total Matches | Correct matches by performance level FBB | BB | B | P | A |
|---|---|---|---|---|---|---|---|
| 2 | 0.574 | 0.61 | 0.642 | 0.614 | 0.452 | 0.532 | 0.703 |
| 3 | 0.588 | 0.62 | 0.658 | 0.625 | 0.5 | 0.536 | 0.728 |
| 4 | 0.603 | 0.633 | 0.629 | 0.627 | 0.571 | 0.582 | 0.642 |
| 5 | 0.555 | 0.595 | 0.509 | 0.493 | 0.492 | 0.639 | 0.749 |
| 6 | 0.577 | 0.612 | 0.326 | 0.536 | 0.556 | 0.626 | 0.762 |
| 7 | 0.541 | 0.584 | 0.449 | 0.453 | 0.542 | 0.644 | 0.687 |

**Table 6.**
**ELA CST 2003 Test-retest Accuracy: overall and by performance level**

| Grade | Average Correct Matches | Average Total Matches | Correct matches by performance level FBB | BB | B | P | A |
|---|---|---|---|---|---|---|---|
| 2 | 0.582 | 0.615 | 0.665 | 0.476 | 0.606 | 0.566 | 0.648 |
| 3 | 0.577 | 0.612 | 0.715 | 0.501 | 0.586 | 0.552 | 0.571 |
| 4 | 0.592 | 0.625 | 0.48 | 0.476 | 0.644 | 0.553 | 0.714 |
| 5 | 0.567 | 0.606 | 0.48 | 0.398 | 0.628 | 0.601 | 0.649 |
| 6 | 0.594 | 0.626 | 0.645 | 0.437 | 0.639 | 0.565 | 0.686 |
| 7 | 0.572 | 0.608 | 0.626 | 0.43 | 0.604 | 0.6 | 0.615 |
| 8 | 0.555 | 0.593 | 0.652 | 0.45 | 0.597 | 0.518 | 0.615 |
| 9 | 0.57 | 0.606 | 0.607 | 0.474 | 0.604 | 0.545 | 0.651 |
| 10 | 0.561 | 0.602 | 0.546 | 0.49 | 0.64 | 0.541 | 0.552 |
| 11 | 0.57 | 0.607 | 0.727 | 0.46 | 0.607 | 0.487 | 0.577 |

**Table 7.**
**Math CST 2002 Test-retest Accuracy: overall and by performance level**

| Grade | Average Correct Matches | Average Total Matches | Correct matches by performance level FBB | BB | B | P | A |
|---|---|---|---|---|---|---|---|
| 2 | 0.583 | 0.616 | 0.673 | 0.595 | 0.498 | 0.559 | 0.719 |
| 3 | 0.564 | 0.6 | 0.662 | 0.599 | 0.487 | 0.552 | 0.628 |
| 4 | 0.589 | 0.622 | 0.53 | 0.655 | 0.585 | 0.531 | 0.605 |
| 5 | 0.51 | 0.56 | 0.354 | 0.472 | 0.495 | 0.579 | 0.696 |
| 6 | 0.538 | 0.581 | 0.349 | 0.497 | 0.525 | 0.583 | 0.723 |
| 7 | 0.515 | 0.563 | 0.444 | 0.465 | 0.51 | 0.592 | 0.62 |

**Table 8.**
**ELA CST 2002 Test-retest Accuracy: overall and by performance level**

| Grade | Average Correct Matches | Average Total Matches | Correct matches by performance level FBB | BB | B | P | A |
|---|---|---|---|---|---|---|---|
| 2 | 0.574 | 0.61 | 0.654 | 0.499 | 0.62 | 0.553 | 0.524 |
| 3 | 0.598 | 0.629 | 0.691 | 0.535 | 0.602 | 0.58 | 0.635 |
| 4 | 0.606 | 0.637 | 0.564 | 0.475 | 0.677 | 0.583 | 0.687 |
| 5 | 0.573 | 0.61 | 0.53 | 0.427 | 0.644 | 0.584 | 0.646 |
| 6 | 0.595 | 0.626 | 0.713 | 0.466 | 0.638 | 0.563 | 0.587 |
| 7 | 0.594 | 0.626 | 0.721 | 0.5 | 0.604 | 0.595 | 0.551 |
| 8 | 0.59 | 0.623 | 0.708 | 0.448 | 0.65 | 0.511 | 0.667 |
| 9 | 0.606 | 0.636 | 0.712 | 0.48 | 0.644 | 0.543 | 0.672 |
| 10 | 0.59 | 0.623 | 0.651 | 0.506 | 0.621 | 0.541 | 0.668 |
| 11 | 0.574 | 0.611 | 0.718 | 0.463 | 0.614 | 0.486 | 0.625 |

**Table 9.**
**Math CST 2003 Proficiency Standard Accuracy**

| | | Missclassification Matrices | | Average Hitrate | Reversal Probability |
|---|---|---|---|---|---|
| | | notP | P | | |
| grade 2 | | | | | |
| | notP | 0.9065 | 0.0935 | | |
| | P | 0.0728 | 0.9272 | 0.9173 | 0.0068 |
| grade 3 | | | | | |
| | notP | 0.929 | 0.071 | | |
| | P | 0.0858 | 0.9142 | 0.9223 | 0.0061 |
| grade 4 | | | | | |
| | notP | 0.9311 | 0.0689 | | |
| | P | 0.0686 | 0.9314 | 0.9312 | 0.0047 |
| grade 5 | | | | | |
| | notP | 0.937 | 0.063 | | |
| | P | 0.0835 | 0.9165 | 0.93 | 0.0053 |
| grade 6 | | | | | |
| | notP | 0.9409 | 0.0591 | | |
| | P | 0.0858 | 0.9142 | 0.9319 | 0.0051 |
| grade 7 | | | | | |
| | notP | 0.9427 | 0.0573 | | |
| | P | 0.0954 | 0.9046 | 0.9317 | 0.0055 |

Proficiency Standard False-Positive, False-Negative Details: Probabilities of incorrect observed classification

| | False Positive | | | False Negative | | |
|---|---|---|---|---|---|---|
| Grade | overall | BB/B | mid-B | overall | mid-P | P/A |
| 2 | 0.0935 | 0.0016 | 0.0867 | 0.0728 | 0.0808 | 0.0077 |
| 3 | 0.071 | 0.0016 | 0.0808 | 0.0858 | 0.0622 | 0.0031 |
| 4 | 0.0689 | 0.0002 | 0.0455 | 0.0686 | 0.0766 | 0.0062 |
| 5 | 0.063 | 0.0042 | 0.0986 | 0.0835 | 0.03 | 0.0003 |
| 6 | 0.0591 | 0.0004 | 0.057 | 0.0858 | 0.0377 | 0.0007 |
| 7 | 0.0573 | 0.0009 | 0.0629 | 0.0954 | 0.0401 | 0.0008 |

## Table 10.
## ELA CST 2003 Proficiency Standard Accuracy

| | | Missclassification Matrices | | Average Hitrate | Reversal Probability |
|---|---|---|---|---|---|
| | | notP | P | | |
| grade 2 | | | | | |
| | notP | 0.9396 | 0.0604 | | |
| | P | 0.0887 | 0.9113 | 0.9293 | 0.0054 |
| grade 3 | | | | | |
| | notP | 0.9436 | 0.0564 | | |
| | P | 0.0952 | 0.9048 | 0.9308 | 0.0054 |
| grade 4 | | | | | |
| | notP | 0.9392 | 0.0608 | | |
| | P | 0.0842 | 0.9158 | 0.9301 | 0.0051 |
| grade 5 | | | | | |
| | notP | 0.9405 | 0.0595 | | |
| | P | 0.0933 | 0.9067 | 0.9284 | 0.0056 |
| grade 6 | | | | | |
| | notP | 0.9409 | 0.0591 | | |
| | P | 0.0898 | 0.9102 | 0.93 | 0.0053 |
| grade 7 | | | | | |
| | notP | 0.936 | 0.064 | | |
| | P | 0.0936 | 0.9064 | 0.9255 | 0.006 |
| grade 8 | | | | | |
| | notP | 0.9429 | 0.0571 | | |
| | P | 0.1264 | 0.8736 | 0.9213 | 0.0072 |
| grade 9 | | | | | |
| | notP | 0.9447 | 0.0553 | | |
| | P | 0.0894 | 0.9106 | 0.9316 | 0.0049 |
| grade 10 | | | | | |
| | notP | 0.9539 | 0.0461 | | |
| | P | 0.0952 | 0.9048 | 0.9372 | 0.0044 |
| grade 11 | | | | | |
| | notP | 0.9505 | 0.0495 | | |
| | P | 0.1088 | 0.8912 | 0.931 | 0.0054 |

Proficiency Standard False-Positive, False-Negative Details: Probabilities of incorrect observed classification

| | False Positive | | | False Negative | | |
|---|---|---|---|---|---|---|
| Grade | overall | BB/B | mid-B | overall | mid-P | P/A |
| 2 | 0.0604 | 0.0001 | 0.0348 | 0.0887 | 0.0555 | 0.0023 |
| 3 | 0.0564 | 0.0001 | 0.0455 | 0.0952 | 0.084 | 0.0076 |
| 4 | 0.0608 | 0. | 0.0206 | 0.0842 | 0.0531 | 0.0013 |
| 5 | 0.0595 | 0.0001 | 0.0297 | 0.0933 | 0.0516 | 0.0008 |
| 6 | 0.0591 | 0.0001 | 0.0297 | 0.0898 | 0.0623 | 0.0021 |
| 7 | 0.064 | 0.0001 | 0.0348 | 0.0936 | 0.0423 | 0.0004 |
| 8 | 0.0571 | 0.0002 | 0.0401 | 0.1264 | 0.0779 | 0.0048 |
| 9 | 0.0553 | 0.0001 | 0.0348 | 0.0894 | 0.0689 | 0.0034 |
| 10 | 0.0461 | 0.0001 | 0.0348 | 0.0952 | 0.0859 | 0.0052 |
| 11 | 0.0495 | 0.0004 | 0.0512 | 0.1088 | 0.0863 | 0.0062 |

**Table 11.**
**Math CST 2002 Proficiency Standard Accuracy**

| | | Missclassification Matrices | | Average Hitrate | Reversal Probability |
|---|---|---|---|---|---|
| | | notP | P | | |
| grade 2 | | | | | |
| | notP | 0.9251 | 0.0749 | | |
| | P | 0.0792 | 0.9208 | 0.9233 | 0.0059 |
| grade 3 | | | | | |
| | notP | 0.9293 | 0.0707 | | |
| | notP | 0.0975 | 0.9025 | 0.9193 | 0.0069 |
| grade 4 | | | | | |
| | notP | 0.945 | 0.055 | | |
| | notP | 0.0975 | 0.9025 | 0.929 | 0.0054 |
| grade 5 | | | | | |
| | notP | 0.9433 | 0.0567 | | |
| | notP | 0.119 | 0.881 | 0.9251 | 0.0068 |
| grade 6 | | | | | |
| | notP | 0.938 | 0.062 | | |
| | notP | 0.0888 | 0.9112 | 0.9296 | 0.0055 |
| grade 7 | | | | | |
| | notP | 0.9428 | 0.0572 | | |
| | notP | 0.1232 | 0.8768 | 0.9238 | 0.007 |

## Table 12.
## ELA CST 2002 Proficiency Standard Accuracy

| | | Missclassification Matrices notP | P | Average Hitrate | Reversal Probability |
|---|---|---|---|---|---|
| **grade 2** | | | | | |
| | **notP** | 0.9502 | 0.0498 | | |
| | **P** | 0.0969 | 0.9031 | 0.9293 | 0.0048 |
| **grade 3** | | | | | |
| | **notP** | 0.9507 | 0.0493 | | |
| | **P** | 0.085 | 0.915 | 0.9308 | 0.0042 |
| **grade 4** | | | | | |
| | **notP** | 0.9509 | 0.0491 | | |
| | **P** | 0.0852 | 0.9148 | 0.9301 | 0.0042 |
| **grade 5** | | | | | |
| | **notP** | 0.9466 | 0.0534 | | |
| | **P** | 0.0976 | 0.9024 | 0.9284 | 0.0052 |
| **grade 6** | | | | | |
| | **notP** | 0.9484 | 0.0516 | | |
| | **P** | 0.1096 | 0.8904 | 0.93 | 0.0057 |
| **grade 7** | | | | | |
| | **notP** | 0.9447 | 0.0553 | | |
| | **P** | 0.1035 | 0.8965 | 0.9255 | 0.0057 |
| **grade 8** | | | | | |
| | **notP** | 0.9443 | 0.0557 | | |
| | **P** | 0.105 | 0.895 | 0.9213 | 0.0058 |
| **grade 9** | | | | | |
| | **notP** | 0.9551 | 0.0449 | | |
| | **P** | 0.0963 | 0.9037 | 0.9316 | 0.0043 |
| **grade 10** | | | | | |
| | **notP** | 0.9469 | 0.0531 | | |
| | **P** | 0.0778 | 0.9222 | 0.9372 | 0.0041 |
| **grade 11** | | | | | |
| | **notP** | 0.9515 | 0.0485 | | |
| | **P** | 0.109 | 0.891 | 0.931 | 0.0053 |

## Table 13.
## Math CST 2002, 2003: Detection of Year-to-year Improvement

Improvement 2002-2003

| True 2002 category | Probability that reported category scores show correct categories | increase | decrease |
|---|---|---|---|
| Grade 2 | | | |
| FBB | 0.648 | 0.773 | 0.015 |
| BB | 0.545 | 0.771 | 0.02 |
| B | 0.517 | 0.756 | 0.024 |
| P | 0.638 | 0.76 | 0.019 |
| Grade 3 | | | |
| FBB | 0.645 | 0.761 | 0.017 |
| BB | 0.585 | 0.799 | 0.015 |
| B | 0.533 | 0.779 | 0.019 |
| P | 0.595 | 0.738 | 0.023 |
| Grade 4 | | | |
| FBB | 0.511 | 0.68 | 0.035 |
| BB | 0.568 | 0.792 | 0.016 |
| B | 0.612 | 0.809 | 0.013 |
| P | 0.631 | 0.778 | 0.017 |
| Grade 5 | | | |
| FBB | 0.436 | 0.581 | 0.051 |
| BB | 0.512 | 0.781 | 0.02 |
| B | 0.557 | 0.798 | 0.015 |
| P | 0.664 | 0.818 | 0.011 |
| Grade 6 | | | |
| FBB | 0.398 | 0.552 | 0.072 |
| BB | 0.519 | 0.773 | 0.022 |
| B | 0.581 | 0.785 | 0.017 |
| P | 0.633 | 0.759 | 0.019 |

Table 13 continued

Improvement 2003-2004* (*assuming 2003 properties represent the 2004 testing)

| **True 2003 category** | **Probability that reported category scores show** | | |
|---|---|---|---|
| | **correct categories** | **increase** | **decrease** |
| Grade 2 | | | |
| FBB | 0.633 | 0.758 | 0.017 |
| BB | 0.554 | 0.762 | 0.022 |
| B | 0.492 | 0.743 | 0.027 |
| P | 0.622 | 0.756 | 0.02 |
| Grade 3 | | | |
| FBB | 0.642 | 0.759 | 0.017 |
| BB | 0.598 | 0.805 | 0.014 |
| B | 0.54 | 0.787 | 0.018 |
| P | 0.587 | 0.728 | 0.025 |
| Grade 4 | | | |
| FBB | 0.557 | 0.726 | 0.027 |
| BB | 0.555 | 0.787 | 0.017 |
| B | 0.604 | 0.791 | 0.016 |
| P | 0.66 | 0.778 | 0.016 |
| Grade 5 | | | |
| FBB | 0.523 | 0.668 | 0.036 |
| BB | 0.524 | 0.763 | 0.022 |
| B | 0.555 | 0.777 | 0.018 |
| P | 0.698 | 0.813 | 0.011 |
| Grade 6 | | | |
| FBB | 0.384 | 0.538 | 0.075 |
| BB | 0.539 | 0.764 | 0.022 |
| B | 0.598 | 0.789 | 0.016 |
| P | 0.656 | 0.774 | 0.016 |

## Table 14.
## ELA CST 2002, 2003: Detection of Year-to-year Improvement

Improvement 2002-2003

| **True 2002 category** | **Probability that reported category scores show** correct categories | increase | decrease |
|---|---|---|---|
| Grade 2 | | | |
| FBB | 0.573 | 0.72 | 0.028 |
| BB | 0.541 | 0.792 | 0.017 |
| B | 0.585 | 0.794 | 0.015 |
| P | 0.562 | 0.692 | 0.031 |
| Grade 3 | | | |
| FBB | 0.573 | 0.741 | 0.024 |
| BB | 0.587 | 0.807 | 0.014 |
| B | 0.577 | 0.792 | 0.016 |
| P | 0.643 | 0.767 | 0.018 |
| Grade 4 | | | |
| FBB | 0.474 | 0.665 | 0.045 |
| BB | 0.546 | 0.78 | 0.018 |
| B | 0.638 | 0.816 | 0.011 |
| P | 0.615 | 0.741 | 0.022 |
| Grade 5 | | | |
| FBB | 0.481 | 0.659 | 0.045 |
| BB | 0.522 | 0.774 | 0.018 |
| B | 0.603 | 0.802 | 0.014 |
| P | 0.633 | 0.768 | 0.017 |
| Grade 6 | | | |
| FBB | 0.554 | 0.745 | 0.024 |
| BB | 0.53 | 0.794 | 0.016 |
| B | 0.619 | 0.811 | 0.012 |
| P | 0.589 | 0.726 | 0.024 |
| Grade 7 | | | |
| FBB | 0.57 | 0.744 | 0.024 |
| BB | 0.546 | 0.772 | 0.02 |
| B | 0.559 | 0.767 | 0.02 |
| P | 0.605 | 0.737 | 0.021 |
| Grade 8 | | | |
| FBB | 0.579 | 0.73 | 0.026 |
| BB | 0.521 | 0.758 | 0.022 |
| B | 0.595 | 0.793 | 0.015 |
| P | 0.576 | 0.726 | 0.026 |
| Grade 9 | | | |
| FBB | 0.591 | 0.749 | 0.022 |
| BB | 0.555 | 0.791 | 0.016 |
| B | 0.59 | 0.796 | 0.015 |
| P | 0.548 | 0.691 | 0.031 |
| Grade 10 | | | |
| FBB | 0.547 | 0.709 | 0.031 |
| BB | 0.554 | 0.792 | 0.016 |
| B | 0.55 | 0.77 | 0.019 |
| P | 0.559 | 0.68 | 0.035 |

Table 14 continued

Improvement 2003-2004* (*assuming 2003 properties represent the 2004 testing)

| **True 2003 category** | **Probability that reported category scores show correct categories** | **increase** | **decrease** |
|---|---|---|---|
| Grade 2 | | | |
| FBB | 0.577 | 0.724 | 0.027 |
| BB | 0.528 | 0.784 | 0.018 |
| B | 0.579 | 0.789 | 0.016 |
| P | 0.568 | 0.698 | 0.029 |
| Grade 3 | | | |
| FBB | 0.583 | 0.751 | 0.022 |
| BB | 0.568 | 0.799 | 0.015 |
| B | 0.569 | 0.789 | 0.017 |
| P | 0.628 | 0.767 | 0.018 |
| Grade 4 | | | |
| FBB | 0.437 | 0.629 | 0.056 |
| BB | 0.546 | 0.777 | 0.019 |
| B | 0.622 | 0.809 | 0.013 |
| P | 0.599 | 0.736 | 0.023 |
| Grade 5 | | | |
| FBB | 0.458 | 0.635 | 0.053 |
| BB | 0.504 | 0.767 | 0.019 |
| B | 0.596 | 0.799 | 0.014 |
| P | 0.642 | 0.767 | 0.017 |
| Grade 6 | | | |
| FBB | 0.527 | 0.718 | 0.031 |
| BB | 0.514 | 0.776 | 0.019 |
| B | 0.619 | 0.806 | 0.013 |
| P | 0.59 | 0.728 | 0.024 |
| Grade 7 | | | |
| FBB | 0.531 | 0.705 | 0.034 |
| BB | 0.507 | 0.753 | 0.023 |
| B | 0.559 | 0.764 | 0.02 |
| P | 0.607 | 0.734 | 0.022 |
| Grade 8 | | | |
| FBB | 0.556 | 0.707 | 0.032 |
| BB | 0.521 | 0.768 | 0.02 |
| B | 0.571 | 0.795 | 0.016 |
| P | 0.58 | 0.747 | 0.022 |
| Grade 9 | | | |
| FBB | 0.545 | 0.704 | 0.032 |
| BB | 0.551 | 0.797 | 0.015 |
| B | 0.572 | 0.791 | 0.016 |
| P | 0.549 | 0.688 | 0.032 |
| Grade 10 | | | |
| FBB | 0.501 | 0.663 | 0.042 |
| BB | 0.546 | 0.785 | 0.017 |
| B | 0.559 | 0.783 | 0.017 |
| P | 0.559 | 0.704 | 0.029 |

# Appendices

## Appendix A-D, Test Archives

For each of the four tests, Math 2003, 2002 and ELA 2003, 2002 there are two components of the test archives. The first component is a stacked series of 5x5 matrices. For math, there are 6 5x5 matrices (grades 2-7) and for ELA there are 10 5x5 matrices (grades 2-11). Each 5x5 matrix structure: rows indicate true category membership and columns indicate reported category membership. Elements of the 5x5 matrices are P{true category = i and observed category = j}. For the 2003 tests these are the matrices reported by ETS in Appendix 5.F in the 2003 CST technical Report; ETS gives similar but not identical values for the unconditional 5x5 matrices. As noted in the text the trace of the 5x5 matrix (which ETS terms proportion correctly classified) is the average hit-rate (see tables 1-4, figures 1-4). These matrices are presented in this stacked form for convenient cut-and-paste for any readers wanting to do their own calculations (replicating or extending what is done in this report).

The second component of Appendices A-D are the grade by grade misclassification matrices with elements the conditional probability:

P{true category = i | observed category = j}.

## Appendix E Computational program function, two versions

Appendix E provides the code and documentation for the computational routines (in S) used to obtain the unconditional matrices in App. A-D.

Two versions are given differening only in the input required: individual test responses or the discrete pdf of the test responses.

# Appendix A:  Archive Math CST 2003

## Unconditional 5x5 matrices, stacked across grades

| | | | | |
|---|---|---|---|---|
| 0.03126 | 0.00776 | 0.00000 | 0.00000 | 0.00000 |
| 0.01412 | 0.15632 | 0.02888 | 0.00025 | 0.00000 |
| 0.00000 | 0.03422 | 0.16161 | 0.04441 | 0.00014 |
| 0.00000 | 0.00018 | 0.03770 | 0.20651 | 0.03885 |
| 0.00000 | 0.00000 | 0.00004 | 0.03841 | 0.19936 |
| 0.04602 | 0.01073 | 0.00000 | 0.00000 | 0.00000 |
| 0.01917 | 0.17931 | 0.02827 | 0.00007 | 0.00000 |
| 0.00000 | 0.03810 | 0.18554 | 0.03861 | 0.00006 |
| 0.00000 | 0.00008 | 0.03887 | 0.20162 | 0.03482 |
| 0.00000 | 0.00000 | 0.00002 | 0.02618 | 0.15253 |
| 0.04458 | 0.01162 | 0.00000 | 0.00000 | 0.00000 |
| 0.01921 | 0.16683 | 0.02461 | 0.00001 | 0.00000 |
| 0.00000 | 0.03097 | 0.21290 | 0.03775 | 0.00004 |
| 0.00000 | 0.00000 | 0.03094 | 0.19988 | 0.03115 |
| 0.00000 | 0.00000 | 0.00001 | 0.03766 | 0.15181 |
| 0.07970 | 0.03187 | 0.00011 | 0.00000 | 0.00000 |
| 0.03603 | 0.19674 | 0.04714 | 0.00019 | 0.00000 |
| 0.00011 | 0.03831 | 0.18711 | 0.04130 | 0.00000 |
| 0.00000 | 0.00008 | 0.02841 | 0.19871 | 0.02132 |
| 0.00000 | 0.00000 | 0.00000 | 0.01249 | 0.08038 |
| 0.02280 | 0.01711 | 0.00003 | 0.00000 | 0.00000 |
| 0.03846 | 0.23003 | 0.04563 | 0.00003 | 0.00000 |
| 0.00004 | 0.03934 | 0.22986 | 0.03910 | 0.00000 |
| 0.00000 | 0.00001 | 0.02896 | 0.19444 | 0.02230 |
| 0.00000 | 0.00000 | 0.00000 | 0.01166 | 0.08018 |
| 0.05243 | 0.02565 | 0.00020 | 0.00000 | 0.00000 |
| 0.04951 | 0.19240 | 0.04406 | 0.00002 | 0.00000 |
| 0.00050 | 0.05032 | 0.25533 | 0.04076 | 0.00000 |
| 0.00000 | 0.00001 | 0.02753 | 0.17662 | 0.01600 |
| 0.00000 | 0.00000 | 0.00000 | 0.01173 | 0.05692 |

## Misclassification Matrices: Performance Levels

| **True category** | **Reported Category** FBB | BB | B | P | A |
|---|---|---|---|---|---|
| Grade 2 | | | | | |
| FBB | 0.8011 | 0.1989 | 0. | 0. | 0. |
| BB | 0.0708 | 0.7833 | 0.1447 | 0.0013 | 0. |
| B | 0. | 0.1424 | 0.6723 | 0.1847 | 0.0006 |
| P | 0. | 0.0006 | 0.1331 | 0.7291 | 0.1372 |
| A | 0. | 0. | 0.0002 | 0.1615 | 0.8383 |
| Grade 3 | | | | | |
| FBB | 0.8109 | 0.1891 | 0. | 0. | 0. |
| BB | 0.0845 | 0.7905 | 0.1246 | 0.0003 | 0. |
| B | 0. | 0.1452 | 0.7073 | 0.1472 | 0.0002 |
| P | 0. | 0.0003 | 0.1411 | 0.7321 | 0.1264 |
| A | 0. | 0. | 0.0001 | 0.1465 | 0.8534 |
| Grade 4 | | | | | |
| FBB | 0.7932 | 0.2068 | 0. | 0. | 0. |
| BB | 0.0912 | 0.7919 | 0.1168 | 0. | 0. |
| B | 0. | 0.11 | 0.7559 | 0.134 | 0.0001 |
| P | 0. | 0. | 0.1181 | 0.763 | 0.1189 |
| A | 0. | 0. | 0.0001 | 0.1988 | 0.8012 |
| Grade 5 | | | | | |
| FBB | 0.7136 | 0.2854 | 0.001 | 0. | 0. |
| BB | 0.1286 | 0.7024 | 0.1683 | 0.0007 | 0. |
| B | 0.0004 | 0.1436 | 0.7012 | 0.1548 | 0. |
| P | 0. | 0.0003 | 0.1143 | 0.7996 | 0.0858 |
| A | 0. | 0. | 0. | 0.1345 | 0.8655 |
| Grade 6 | | | | | |
| FBB | 0.5709 | 0.4284 | 0.0008 | 0. | 0. |
| BB | 0.1224 | 0.7322 | 0.1452 | 0.0001 | 0. |
| B | 0.0001 | 0.1276 | 0.7455 | 0.1268 | 0. |
| P | 0. | 0. | 0.1179 | 0.7913 | 0.0908 |
| A | 0. | 0. | 0. | 0.127 | 0.873 |
| Grade 7 | | | | | |
| FBB | 0.6698 | 0.3277 | 0.0026 | 0. | 0. |
| BB | 0.1731 | 0.6728 | 0.1541 | 0.0001 | 0. |
| B | 0.0014 | 0.1451 | 0.736 | 0.1175 | 0. |
| P | 0. | 0. | 0.125 | 0.8022 | 0.0727 |
| A | 0. | 0. | 0. | 0.1709 | 0.8291 |

# Appendix B: Archive Math CST 2002

Unconditional 5x5 matrices, stacked across grades

| | | | | |
|---|---|---|---|---|
| 0.05647 | 0.01237 | 0.00000 | 0.00000 | 0.00000 |
| 0.02266 | 0.19122 | 0.03397 | 0.00010 | 0.00000 |
| 0.00000 | 0.03422 | 0.18620 | 0.04333 | 0.00007 |
| 0.00000 | 0.00005 | 0.03317 | 0.20408 | 0.03566 |
| 0.00000 | 0.00000 | 0.00001 | 0.02226 | 0.12417 |
| 0.06558 | 0.01500 | 0.00000 | 0.00000 | 0.00000 |
| 0.02467 | 0.20558 | 0.03533 | 0.00015 | 0.00000 |
| 0.00000 | 0.04070 | 0.19633 | 0.04413 | 0.00007 |
| 0.00000 | 0.00010 | 0.03621 | 0.18853 | 0.02900 |
| 0.00000 | 0.00000 | 0.00001 | 0.02462 | 0.09398 |
| 0.04525 | 0.01692 | 0.00000 | 0.00000 | 0.00000 |
| 0.02108 | 0.20831 | 0.02791 | 0.00001 | 0.00000 |
| 0.00000 | 0.03702 | 0.23183 | 0.03416 | 0.00007 |
| 0.00000 | 0.00001 | 0.03678 | 0.18234 | 0.03107 |
| 0.00000 | 0.00000 | 0.00002 | 0.02828 | 0.09896 |
| 0.03883 | 0.02628 | 0.00016 | 0.00000 | 0.00000 |
| 0.05475 | 0.23146 | 0.05051 | 0.00016 | 0.00000 |
| 0.00025 | 0.05057 | 0.21572 | 0.04005 | 0.00001 |
| 0.00000 | 0.00012 | 0.03453 | 0.17137 | 0.01927 |
| 0.00000 | 0.00000 | 0.00000 | 0.01094 | 0.05502 |
| 0.03138 | 0.02162 | 0.00008 | 0.00000 | 0.00000 |
| 0.04908 | 0.22293 | 0.04420 | 0.00005 | 0.00000 |
| 0.00014 | 0.04501 | 0.23087 | 0.04260 | 0.00003 |
| 0.00000 | 0.00002 | 0.02768 | 0.16763 | 0.02424 |
| 0.00000 | 0.00000 | 0.00000 | 0.01383 | 0.07861 |
| 0.05927 | 0.02947 | 0.00018 | 0.00000 | 0.00000 |
| 0.05068 | 0.20913 | 0.04687 | 0.00007 | 0.00000 |
| 0.00034 | 0.04945 | 0.22578 | 0.04066 | 0.00002 |
| 0.00000 | 0.00006 | 0.03542 | 0.18428 | 0.01971 |
| 0.00000 | 0.00000 | 0.00000 | 0.01034 | 0.03826 |

## Misclassification Matrices: Performance Levels

| **True** | **Reported Category** | | | | |
|---|---|---|---|---|---|
| **category** | FBB | BB | B | P | A |
| Grade 2 | | | | | |
| FBB | 0.8203 | 0.1797 | 0. | 0. | 0. |
| BB | 0.0914 | 0.7712 | 0.137 | 0.0004 | 0. |
| B | 0. | 0.1297 | 0.7058 | 0.1642 | 0.0003 |
| P | 0. | 0.0002 | 0.1215 | 0.7477 | 0.1306 |
| A | 0. | 0. | 0.0001 | 0.152 | 0.8479 |
| Grade 3 | | | | | |
| FBB | 0.8138 | 0.1862 | 0. | 0. | 0. |
| BB | 0.0928 | 0.7736 | 0.133 | 0.0006 | 0. |
| B | 0. | 0.1447 | 0.6981 | 0.1569 | 0.0002 |
| P | 0. | 0.0004 | 0.1426 | 0.7427 | 0.1142 |
| A | 0. | 0. | 0.0001 | 0.2076 | 0.7923 |
| Grade 4 | | | | | |
| FBB | 0.7278 | 0.2722 | 0. | 0. | 0. |
| BB | 0.0819 | 0.8096 | 0.1085 | 0. | 0. |
| B | 0. | 0.1221 | 0.7649 | 0.1127 | 0.0002 |
| P | 0. | 0. | 0.147 | 0.7288 | 0.1242 |
| A | 0. | 0. | 0.0002 | 0.2222 | 0.7776 |
| Grade 5 | | | | | |
| FBB | 0.5949 | 0.4026 | 0.0025 | 0. | 0. |
| BB | 0.1625 | 0.6871 | 0.1499 | 0.0005 | 0. |
| B | 0.0008 | 0.1649 | 0.7036 | 0.1306 | 0. |
| P | 0. | 0.0005 | 0.1533 | 0.7607 | 0.0855 |
| A | 0. | 0. | 0. | 0.1659 | 0.8341 |
| Grade 6 | | | | | |
| FBB | 0.5912 | 0.4073 | 0.0015 | 0. | 0. |
| BB | 0.1552 | 0.7049 | 0.1398 | 0.0002 | 0. |
| B | 0.0004 | 0.1413 | 0.7245 | 0.1337 | 0.0001 |
| P | 0. | 0.0001 | 0.1261 | 0.7634 | 0.1104 |
| A | 0. | 0. | 0. | 0.1496 | 0.8504 |
| Grade 7 | | | | | |
| FBB | 0.6666 | 0.3314 | 0.002 | 0. | 0. |
| BB | 0.1652 | 0.6818 | 0.1528 | 0.0002 | 0. |
| B | 0.0011 | 0.1564 | 0.7139 | 0.1286 | 0.0001 |
| P | 0. | 0.0003 | 0.1479 | 0.7695 | 0.0823 |
| A | 0. | 0. | 0. | 0.2128 | 0.7872 |

# Appendix C:  Archive ELA CST 2003

Unconditional 5x5 matrices, stacked across grades

| | | | | |
|---|---|---|---|---|
| 0.08770 | 0.01978 | 0.00005 | 0.00000 | 0.00000 |
| 0.03040 | 0.13662 | 0.03095 | 0.00000 | 0.00000 |
| 0.00013 | 0.03508 | 0.25987 | 0.03860 | 0.00004 |
| 0.00000 | 0.00000 | 0.03200 | 0.18490 | 0.02888 |
| 0.00000 | 0.00000 | 0.00001 | 0.02239 | 0.09260 |
| 0.12314 | 0.02249 | 0.00003 | 0.00000 | 0.00000 |
| 0.03072 | 0.14998 | 0.03115 | 0.00001 | 0.00000 |
| 0.00006 | 0.03570 | 0.24016 | 0.03777 | 0.00006 |
| 0.00000 | 0.00001 | 0.03128 | 0.16709 | 0.02651 |
| 0.00000 | 0.00000 | 0.00001 | 0.02539 | 0.07843 |
| 0.04056 | 0.01788 | 0.00010 | 0.00000 | 0.00000 |
| 0.02618 | 0.12676 | 0.03087 | 0.00000 | 0.00000 |
| 0.00022 | 0.03534 | 0.29490 | 0.03702 | 0.00005 |
| 0.00000 | 0.00000 | 0.03284 | 0.17828 | 0.02855 |
| 0.00000 | 0.00000 | 0.00002 | 0.02333 | 0.12709 |
| 0.05926 | 0.02578 | 0.00048 | 0.00000 | 0.00000 |
| 0.03305 | 0.11765 | 0.03582 | 0.00000 | 0.00000 |
| 0.00069 | 0.03785 | 0.29289 | 0.03817 | 0.00001 |
| 0.00000 | 0.00000 | 0.03343 | 0.20686 | 0.02651 |
| 0.00000 | 0.00000 | 0.00000 | 0.01778 | 0.07378 |
| 0.08665 | 0.02107 | 0.00015 | 0.00000 | 0.00000 |
| 0.02707 | 0.11143 | 0.03008 | 0.00000 | 0.00000 |
| 0.00029 | 0.03528 | 0.29347 | 0.03799 | 0.00003 |
| 0.00000 | 0.00000 | 0.03200 | 0.17431 | 0.02555 |
| 0.00000 | 0.00000 | 0.00001 | 0.02142 | 0.10320 |
| 0.08482 | 0.02219 | 0.00018 | 0.00000 | 0.00000 |
| 0.03008 | 0.12909 | 0.03761 | 0.00001 | 0.00000 |
| 0.00027 | 0.03462 | 0.26540 | 0.04132 | 0.00001 |
| 0.00000 | 0.00000 | 0.03317 | 0.20355 | 0.02613 |
| 0.00000 | 0.00000 | 0.00000 | 0.01974 | 0.07179 |
| 0.09791 | 0.02318 | 0.00013 | 0.00000 | 0.00000 |
| 0.03288 | 0.14216 | 0.03687 | 0.00001 | 0.00000 |
| 0.00025 | 0.04116 | 0.27428 | 0.03920 | 0.00008 |
| 0.00000 | 0.00001 | 0.03939 | 0.17035 | 0.02702 |
| 0.00000 | 0.00000 | 0.00002 | 0.01618 | 0.05892 |
| 0.07963 | 0.02250 | 0.00009 | 0.00000 | 0.00000 |
| 0.03228 | 0.13849 | 0.03037 | 0.00000 | 0.00000 |
| 0.00020 | 0.03580 | 0.24513 | 0.03418 | 0.00005 |
| 0.00000 | 0.00000 | 0.03407 | 0.18050 | 0.02984 |
| 0.00000 | 0.00000 | 0.00002 | 0.02643 | 0.11038 |
| 0.10034 | 0.03537 | 0.00011 | 0.00000 | 0.00000 |
| 0.03180 | 0.15769 | 0.03574 | 0.00000 | 0.00000 |
| 0.00009 | 0.02924 | 0.23925 | 0.03037 | 0.00005 |
| 0.00000 | 0.00000 | 0.03231 | 0.16353 | 0.02648 |
| 0.00000 | 0.00000 | 0.00004 | 0.03018 | 0.08741 |
| 0.15433 | 0.02658 | 0.00007 | 0.00000 | 0.00000 |
| 0.03199 | 0.13576 | 0.03237 | 0.00001 | 0.00000 |
| 0.00010 | 0.03048 | 0.22504 | 0.03300 | 0.00014 |
| 0.00000 | 0.00001 | 0.03582 | 0.15456 | 0.03100 |
| 0.00000 | 0.00000 | 0.00008 | 0.02606 | 0.08262 |

## Misclassification Matrices: Performance Levels

| **True** | **Reported Category** | | | | |
|---|---|---|---|---|---|
| **category** | FBB | BB | B | P | A |
| Grade 2 | | | | | |
| FBB | 0.8156 | 0.1839 | 0.0005 | 0. | 0. |
| BB | 0.1536 | 0.6901 | 0.1563 | 0. | 0. |
| B | 0.0004 | 0.1051 | 0.7787 | 0.1157 | 0.0001 |
| P | 0. | 0. | 0.1302 | 0.7523 | 0.1175 |
| A | 0. | 0. | 0.0001 | 0.1947 | 0.8052 |
| Grade 3 | | | | | |
| FBB | 0.8454 | 0.1544 | 0.0002 | 0. | 0. |
| BB | 0.145 | 0.7079 | 0.147 | 0. | 0. |
| B | 0.0002 | 0.1138 | 0.7655 | 0.1204 | 0.0002 |
| P | 0. | 0. | 0.1391 | 0.743 | 0.1179 |
| A | 0. | 0. | 0.0001 | 0.2445 | 0.7554 |
| Grade 4 | | | | | |
| FBB | 0.6929 | 0.3054 | 0.0017 | 0. | 0. |
| BB | 0.1424 | 0.6896 | 0.1679 | 0. | 0. |
| B | 0.0006 | 0.0962 | 0.8024 | 0.1007 | 0.0001 |
| P | 0. | 0. | 0.137 | 0.7439 | 0.1191 |
| A | 0. | 0. | 0.0001 | 0.1551 | 0.8448 |
| Grade 5 | | | | | |
| FBB | 0.6929 | 0.3014 | 0.0056 | 0. | 0. |
| BB | 0.1772 | 0.6308 | 0.192 | 0. | 0. |
| B | 0.0019 | 0.1024 | 0.7924 | 0.1033 | 0. |
| P | 0. | 0. | 0.1253 | 0.7753 | 0.0994 |
| A | 0. | 0. | 0. | 0.1942 | 0.8058 |
| Grade 6 | | | | | |
| FBB | 0.8033 | 0.1953 | 0.0014 | 0. | 0. |
| BB | 0.1606 | 0.661 | 0.1784 | 0. | 0. |
| B | 0.0008 | 0.0961 | 0.7995 | 0.1035 | 0.0001 |
| P | 0. | 0. | 0.138 | 0.7518 | 0.1102 |
| A | 0. | 0. | 0.0001 | 0.1719 | 0.8281 |

## Misclassification Matrices: Performance Levels

| **True category** | **Reported Category** FBB | BB | B | P | A |
|---|---|---|---|---|---|
| Grade 7 | | | | | |
| FBB | 0.7913 | 0.207 | 0.0017 | 0. | 0. |
| BB | 0.1529 | 0.656 | 0.1911 | 0.0001 | 0. |
| B | 0.0008 | 0.1013 | 0.7769 | 0.121 | 0. |
| P | 0. | 0. | 0.1262 | 0.7744 | 0.0994 |
| A | 0. | 0. | 0. | 0.2157 | 0.7843 |
| Grade 8 | | | | | |
| FBB | 0.8077 | 0.1912 | 0.0011 | 0. | 0. |
| BB | 0.1552 | 0.6708 | 0.174 | 0. | 0. |
| B | 0.0007 | 0.116 | 0.7727 | 0.1104 | 0.0002 |
| P | 0. | 0. | 0.1664 | 0.7195 | 0.1141 |
| A | 0. | 0. | 0.0003 | 0.2154 | 0.7843 |
| Grade 9 | | | | | |
| FBB | 0.779 | 0.2201 | 0.0009 | 0. | 0. |
| BB | 0.1605 | 0.6885 | 0.151 | 0. | 0. |
| B | 0.0006 | 0.1135 | 0.7773 | 0.1084 | 0.0002 |
| P | 0. | 0. | 0.1394 | 0.7385 | 0.1221 |
| A | 0. | 0. | 0.0001 | 0.1932 | 0.8067 |
| Grade 10 | | | | | |
| FBB | 0.7388 | 0.2604 | 0.0008 | 0. | 0. |
| BB | 0.1412 | 0.7001 | 0.1587 | 0. | 0. |
| B | 0.0003 | 0.0978 | 0.8002 | 0.1016 | 0.0002 |
| P | 0. | 0. | 0.1453 | 0.7356 | 0.1191 |
| A | 0. | 0. | 0.0003 | 0.2566 | 0.7431 |
| Grade 11 | | | | | |
| FBB | 0.8527 | 0.1469 | 0.0004 | 0. | 0. |
| BB | 0.1598 | 0.6784 | 0.1617 | 0. | 0. |
| B | 0.0003 | 0.1056 | 0.7793 | 0.1143 | 0.0005 |
| P | 0. | 0. | 0.1618 | 0.6981 | 0.14 |
| A | 0. | 0. | 0.0007 | 0.2396 | 0.7597 |

# Appendix D: Archive ELA CST 2002

## Unconditional 5x5 matrices, stacked across grades

| | | | | |
|---|---|---|---|---|
| 0.11825 | 0.02792 | 0.00004 | 0.00000 | 0.00000 |
| 0.03359 | 0.15972 | 0.03287 | 0.00000 | 0.00000 |
| 0.00005 | 0.03258 | 0.24767 | 0.03418 | 0.00004 |
| 0.00000 | 0.00000 | 0.03034 | 0.17323 | 0.02944 |
| 0.00000 | 0.00000 | 0.00001 | 0.02213 | 0.05796 |
| 0.12618 | 0.02564 | 0.00002 | 0.00000 | 0.00000 |
| 0.03050 | 0.16871 | 0.03144 | 0.00000 | 0.00000 |
| 0.00003 | 0.03116 | 0.22261 | 0.03299 | 0.00002 |
| 0.00000 | 0.00000 | 0.02810 | 0.17273 | 0.02606 |
| 0.00000 | 0.00000 | 0.00000 | 0.02107 | 0.08273 |
| 0.07879 | 0.02604 | 0.00012 | 0.00000 | 0.00000 |
| 0.02724 | 0.12880 | 0.03076 | 0.00000 | 0.00000 |
| 0.00014 | 0.03050 | 0.28822 | 0.03154 | 0.00001 |
| 0.00000 | 0.00000 | 0.03050 | 0.18623 | 0.02710 |
| 0.00000 | 0.00000 | 0.00000 | 0.01953 | 0.09449 |
| 0.07476 | 0.02764 | 0.00027 | 0.00000 | 0.00000 |
| 0.03568 | 0.14157 | 0.03938 | 0.00000 | 0.00000 |
| 0.00038 | 0.03827 | 0.31026 | 0.03771 | 0.00002 |
| 0.00000 | 0.00000 | 0.02871 | 0.16173 | 0.02127 |
| 0.00000 | 0.00000 | 0.00000 | 0.01616 | 0.06616 |
| 0.11778 | 0.02163 | 0.00008 | 0.00000 | 0.00000 |
| 0.03143 | 0.13213 | 0.02995 | 0.00000 | 0.00000 |
| 0.00019 | 0.03784 | 0.29528 | 0.03621 | 0.00003 |
| 0.00000 | 0.00000 | 0.03259 | 0.17797 | 0.02655 |
| 0.00000 | 0.00000 | 0.00000 | 0.01410 | 0.04624 |
| 0.13077 | 0.02319 | 0.00004 | 0.00000 | 0.00000 |
| 0.02757 | 0.15014 | 0.03462 | 0.00001 | 0.00000 |
| 0.00006 | 0.03455 | 0.25404 | 0.03835 | 0.00001 |
| 0.00000 | 0.00000 | 0.03173 | 0.18522 | 0.02318 |
| 0.00000 | 0.00000 | 0.00000 | 0.01714 | 0.04937 |
| 0.11521 | 0.02157 | 0.00011 | 0.00000 | 0.00000 |
| 0.02959 | 0.13666 | 0.03782 | 0.00000 | 0.00000 |
| 0.00014 | 0.03178 | 0.29648 | 0.03933 | 0.00014 |
| 0.00000 | 0.00000 | 0.03053 | 0.14619 | 0.02786 |
| 0.00000 | 0.00000 | 0.00003 | 0.01583 | 0.07073 |
| 0.15304 | 0.02828 | 0.00008 | 0.00000 | 0.00000 |
| 0.03016 | 0.13930 | 0.03151 | 0.00000 | 0.00000 |
| 0.00010 | 0.03094 | 0.25329 | 0.03126 | 0.00005 |
| 0.00000 | 0.00000 | 0.02907 | 0.14974 | 0.02433 |
| 0.00000 | 0.00000 | 0.00001 | 0.01778 | 0.08106 |
| 0.12816 | 0.03063 | 0.00004 | 0.00000 | 0.00000 |
| 0.03040 | 0.15548 | 0.03269 | 0.00000 | 0.00000 |
| 0.00004 | 0.02840 | 0.24081 | 0.03615 | 0.00009 |
| 0.00000 | 0.00000 | 0.02464 | 0.14848 | 0.02882 |
| 0.00000 | 0.00000 | 0.00002 | 0.02104 | 0.09411 |
| 0.14927 | 0.02683 | 0.00009 | 0.00000 | 0.00000 |
| 0.03219 | 0.14934 | 0.03796 | 0.00001 | 0.00000 |
| 0.00009 | 0.02992 | 0.22905 | 0.03320 | 0.00016 |
| 0.00000 | 0.00000 | 0.03392 | 0.14105 | 0.02730 |
| 0.00000 | 0.00000 | 0.00008 | 0.02290 | 0.08661 |

## Misclassification Matrices: Performance Levels

| **True** | **Reported Category** | | | | |
|---|---|---|---|---|---|
| **category** | FBB | BB | B | P | A |
| Grade 2 | | | | | |
| FBB | 0.8088 | 0.191 | 0.0003 | 0. | 0. |
| BB | 0.1485 | 0.7062 | 0.1453 | 0. | 0. |
| B | 0.0002 | 0.1036 | 0.7875 | 0.1087 | 0.0001 |
| P | 0. | 0. | 0.1302 | 0.7434 | 0.1263 |
| A | 0. | 0. | 0.0001 | 0.2763 | 0.7236 |
| Grade 3 | | | | | |
| FBB | 0.831 | 0.1689 | 0.0001 | 0. | 0. |
| BB | 0.1322 | 0.7315 | 0.1363 | 0. | 0. |
| B | 0.0001 | 0.1086 | 0.7762 | 0.115 | 0.0001 |
| P | 0. | 0. | 0.1238 | 0.7613 | 0.1149 |
| A | 0. | 0. | 0. | 0.203 | 0.797 |
| Grade 4 | | | | | |
| FBB | 0.7507 | 0.2481 | 0.0011 | 0. | 0. |
| BB | 0.1458 | 0.6895 | 0.1647 | 0. | 0. |
| B | 0.0004 | 0.087 | 0.8225 | 0.09 | 0. |
| P | 0. | 0. | 0.1251 | 0.7638 | 0.1111 |
| A | 0. | 0. | 0. | 0.1713 | 0.8287 |
| Grade 5 | | | | | |
| FBB | 0.7282 | 0.2692 | 0.0026 | 0. | 0. |
| BB | 0.1647 | 0.6535 | 0.1818 | 0. | 0. |
| B | 0.001 | 0.099 | 0.8025 | 0.0975 | 0.0001 |
| P | 0. | 0. | 0.1356 | 0.7639 | 0.1005 |
| A | 0. | 0. | 0. | 0.1963 | 0.8037 |
| Grade 6 | | | | | |
| FBB | 0.8444 | 0.1551 | 0.0006 | 0. | 0. |
| BB | 0.1624 | 0.6828 | 0.1548 | 0. | 0. |
| B | 0.0005 | 0.1024 | 0.799 | 0.098 | 0.0001 |
| P | 0. | 0. | 0.1374 | 0.7506 | 0.112 |
| A | 0. | 0. | 0. | 0.2337 | 0.7663 |

## Misclassification Matrices: Performance Levels

| **True category** | Reported Category FBB | BB | B | P | A |
|---|---|---|---|---|---|
| Grade 7 | | | | | |
| FBB | 0.8492 | 0.1506 | 0.0003 | 0. | 0. |
| BB | 0.1298 | 0.7071 | 0.163 | 0. | 0. |
| B | 0.0002 | 0.1057 | 0.7769 | 0.1173 | 0. |
| P | 0. | 0. | 0.1321 | 0.7713 | 0.0965 |
| A | 0. | 0. | 0. | 0.2577 | 0.7423 |
| Grade 8 | | | | | |
| FBB | 0.8416 | 0.1576 | 0.0008 | 0. | 0. |
| BB | 0.145 | 0.6697 | 0.1853 | 0. | 0. |
| B | 0.0004 | 0.0864 | 0.8059 | 0.1069 | 0.0004 |
| P | 0. | 0. | 0.1492 | 0.7146 | 0.1362 |
| A | 0. | 0. | 0.0003 | 0.1828 | 0.8168 |
| Grade 9 | | | | | |
| FBB | 0.8437 | 0.1559 | 0.0004 | 0. | 0. |
| BB | 0.1501 | 0.6931 | 0.1568 | 0. | 0. |
| B | 0.0003 | 0.098 | 0.8025 | 0.099 | 0.0002 |
| P | 0. | 0. | 0.1431 | 0.7371 | 0.1198 |
| A | 0. | 0. | 0.0001 | 0.1799 | 0.82 |
| Grade 10 | | | | | |
| FBB | 0.8069 | 0.1928 | 0.0003 | 0. | 0. |
| BB | 0.1391 | 0.7114 | 0.1496 | 0. | 0. |
| B | 0.0001 | 0.093 | 0.7883 | 0.1183 | 0.0003 |
| P | 0. | 0. | 0.122 | 0.7353 | 0.1427 |
| A | 0. | 0. | 0.0002 | 0.1827 | 0.8171 |
| Grade 11 | | | | | |
| FBB | 0.8472 | 0.1523 | 0.0005 | 0. | 0. |
| BB | 0.1467 | 0.6804 | 0.1729 | 0. | 0. |
| B | 0.0003 | 0.1023 | 0.7833 | 0.1135 | 0.0005 |
| P | 0. | 0. | 0.1677 | 0.6973 | 0.135 |
| A | 0. | 0. | 0.0007 | 0.209 | 0.7903 |

# Appendix E Computational Procedures

The following SPlus function is called "llclassify". It is an implementation of Livingston and Lewis' (1995) method for estimating the accuracy of classifications based on test scores. We have made a slight modification whereby in estimating the joint distribution of true scores and observed scores, we integrate over the true score distribution rather than dividing it into bins. This modification leads to simpler code, faster output, and greater precision. The main output is a classification matrix, where a row denotes the true score category and a column represents the observed score category.

The function is given below. Comments are denoted by #. The function call is of the form

```
function(observedy, r, max, cut, min = 0, adjust = 1).


# The arguments of this function are as follows:
# First argument is the inputted data. Data should
# be a vector of numeric scores (either scale or
# number correct); that is, if 400,000 students are
# examined, then "observedy" is a vector of 400,000
# numbers representing the score of each student. If
# a pdf file is given instead of a vector, that pdf
# file should be expanded into such a vector. Second
# argument is inputted reliability coefficient. Third
# argument is the maximum possible score on the test.
# Fourth argument is vector of cut points. Cuts are
# the lower bound of the "higher" category. For
# instance, if Far Below Basic stops at a score of
# 26, and Below Basic starts at 27, then put in 27
# for that cut. Fifth argument is minimum possible
# score on the test (default=0). Sixth argument is
# whether to adjust the results to match the
# marginals of the observed score distribution;
# default (1) is to adjust them. Put in a value other
# than 1 if you do not want to adjust the values in
# this way.

{
# Adjust the scores in case min is not 0, and do
# appropriate calculations.
        observed <- observedy - min
        max <- max - min
        cut <- cut - min
        x <- round(observed)
        p <- (x)/(max)
        mup <- sum(p)/length(p)
        deviations <- (p - mup)
        squaredeviations <- deviations^2
        sigma2p <- sum(squaredeviations)
        /(length(p))

# Find L&L's effective test length, and adjust
# observed scores according to effective test
# length.

        effective <- (mup * (1 - mup) - r *
        sigma2p)/(sigma2p * (1 - r))
        n <- round(effective)

# Fit 4-parameter beta distribution to true scores,
# using method of moments.

        xprimenotrounded <- (n * (x))/(max)
        xprime <- round(xprimenotrounded)
        mprime1x <- sum(xprime)/length(xprime)
        mprime2x <- sum(xprime^2)/length(xprime)
        mprime3x <- sum(xprime^3)/length(xprime)
        mprime4x <- sum(xprime^4)/length(xprime)
        mprime1p <- mprime1x/n
        mprime2p <- (mprime2x - mprime1x)/
        (n * (n - 1))
        mprime3p <- (mprime3x - 3 * mprime2x +
        2 * mprime1x)/(n * (n - 1) * (n - 2))
        mprime4p <- (mprime4x - 6 * mprime3x +
        11 * mprime2x - 6 * mprime1x)/
        (n * (n - 1) * (n - 2) * (n - 3))
        m2p <- (mprime2p - mprime1p^2)
```

```
        m3p <- (mprime3p - 3 * mprime1p * mprime2p
        + 2 * mprime1p^3)
        m4p <- (mprime4p - 4 * mprime1p * mprime3p
        + 6 * mprime1p^2 * mprime2p - 3 *
        mprime1p^4)
        k <- (16 * m2p^3)/m3p^2
        l <- (m4p * m2p)/m3p^2
        phi <- (3 * (k - 16 * (l - 1)))/
        (16 * (l - 1) - 8 - 3 * k)
        sgnm3p <- m3p/abs(m3p)

# Revert to 3-parameter beta distribution with
# beta=1 if 4-parameter method of moments does not
# give an appropriate solution.

        if(as.double(k * (phi + 1) + (phi + 2)^2)
        <= 0) {
        z <- (2 * (1 - mprime1p) * m2p)/m3p
        new <- m2p/(1 - mprime1p)^2
        betahat <- (z * (1 - new) - 2)/(1 - new +
        2 * new * z)
        alphahat <- (new * betahat * (1 + betahat))
        /(1 - new * betahat)
        ahat <- ((alphahat + betahat) * mprime1p -
        alphahat)/betahat
        bhat <- 1
        }

# Fit the 4-parameter beta distribution if method of
# moments gives an appropriate solution.  alphahat,
# betahat, ahat, and bhat are estimates of the 4
# respective parameters:  alpha, beta, a, and b.
# alpha and beta are the usual parameters of the
# 2-parameter beta distribution.  a is the lowest
# value that the true score can take, while b is the
# highest such value.

        else {
        theta <- (sgnm3p * phi * (phi + 2))/
        sqrt(k * (phi + 1) + (phi + 2)^2)
        alphahat <- (phi - theta)/2
        betahat <- (phi + theta)/2
        ahat <- (mprime1p - sqrt((m2p * alphahat
        * (alphahat + betahat + 1))/betahat))
        bhat <- (mprime1p + sqrt((m2p * betahat *
        (alphahat + betahat + 1))/alphahat))
        }
        if((ahat < 0) || (alphahat < 0) ||
        (betahat < 0)) {
        rrr <- m2p/mprime1p^2
        w <- (2 * m2p^2)/(mprime1p * m3p)
        alphahat <- (w * (rrr - 1) - 2)/(1 - rrr -
        2 * rrr * w)
        betahat <- (rrr * alphahat * (alphahat + 1))
        /(1 - rrr * alphahat)
        bhat <- ((alphahat + betahat) * mprime1p)
        /alphahat
        ahat <- 0
        }

# Fit a simple beta model if the method of moments
# continues not to give an appropriate solution.

        if((bhat > 1) || (alphahat < 0) || (betahat
        < 0)) {
        z <- (2 * (1 - mprime1p) * m2p)/m3p
        new <- m2p/(1 - mprime1p)^2
        betahat <- (z * (1 - new) - 2)/(1 - new +
        2 * new * z)
        alphahat <- (new * betahat * (1 + betahat))
        /(1 - new * betahat)
        ahat <- ((alphahat + betahat) * mprime1p -
        alphahat)/betahat
        bhat <- 1
        }

# Figure out which values inside [ahat, bhat] belong
# in which true score category.

        cutinp <- (cut - 0.5)/max
        cutinp2 <- c(ahat, cutinp, bhat)

# Figure out which observed scores will map into
# which observed classification.  1e-07 term
# necessary because of unexpected results from
# SPlus' "floor" function.

        cut2 <- floor(((cut - 0.5) * n)/max - 1e-07)
```

```
        cut3 <- c(0, cut2, n)

# Figure out each entry in confusion matrix, which
# gives the joint distribution of true score and
# observed score.  This program integrates over true
# score, rather than L&L's suggestion of dividing
# into bins.  This modification is both more exact
# and much faster.  Note that the additional
# function "ffunc" is required to run this program.
# ffunc is added at the end of this text.

        finalmatrix <- matrix(0, ncol = length(cut)
        + 1, nrow = length(cut) + 1)
        for(i in 1:nrow(finalmatrix)) {
        for(j in 1:nrow(finalmatrix)) {
        finalmatrix[i, j] <- integrate(ffunc,
        cutinp2[i], cutinp2[i + 1], ahat1 = ahat,
        bhat1 = bhat, alphahat1 = alphahat, betahat1
        = betahat, lowbinom = cut3[j], highbinom =
        cut3[j + 1], n1 = n)$integral
        }
        }

# If adjust is set to 1, adjust the entries to match
# the observed score distribution.

        if(adjust == 1) {
        proportions <- c(0, length(cut) + 1)
        xlowest <- x[x < cut[1]]
        proportions[1] <- length(xlowest)/length(x)
        if(length(cut) > 1) {
        for(i in 2:length(cut)) {
        temp <- x[x < cut[i]]
        proportions[i] <- (length(temp)/length(x)
        - sum(proportions[1:(i - 1)]))
        }
        }
        proportions[length(cut) + 1] <- 1 -
        sum(proportions[1:length(cut)])
        for(i in 1:ncol(finalmatrix)) {
        finalmatrix[, i] <- (finalmatrix[, i] *
        proportions[i])/sum(finalmatrix[, i])
        }
        }

# Return the parameters of the fitted 4-parameter
# beta distribution, the effective test length, and
# the confusion matrix.

        list(alphahat = alphahat, betahat = betahat,
        ahat = ahat, bhat = bhat, n = n,
        finalmatrix = finalmatrix)
}



# This short function "ffunc" is needed to perform
# the integral used in the above function.  All
# arguments are defined within the function that
# calls it.

> ffunc
function(x, ahat1, bhat1, alphahat1, betahat1,
lowbinom, highbinom, n1)
{
        (dbeta(((x - ahat1))/(bhat1 - ahat1),
        alphahat1, betahat1) * (pbinom(
        highbinom, n1, x) - pbinom(lowbinom, n1, x)))
        /(bhat1 - ahat1)
}

# The following is an example of the function's
# output.  It is based on 2003 raw score data of
# grade 4 math students.  $alphahat is the
# estimated value of alpha in the 4-parameter beta
# distribution.  $betahat is the estimated value of
# beta in this 4-parameter beta distribution.
# $ahat and $bhat are the estimated lower and upper
# bounds of this true score distribution.  $n is
# the so-called "effective test length" in the
# Livingston-Lewis paper.  $finalmatrix is the final
# confusion matrix outputted by the function.  Its
# entries have been rounded off for ease of viewing.

$alphahat:
[1] 1.377472

$betahat:
[1] 0.783828
```

```
$ahat:
[1] 0.2031712

$bhat:
[1] 0.9505744

$n:
[1] 68

$finalmatrix:

        [,1]    [,2]    [,3]    [,4]    [,5]
[1,] 0.04458 0.01162 0.00000 0.00000 0.00000
[2,] 0.01921 0.16683 0.02461 0.00001 0.00000
[3,] 0.00000 0.03097 0.21290 0.03775 0.00004
[4,] 0.00000 0.00000 0.03094 0.19988 0.03115
[5,] 0.00000 0.00000 0.00001 0.03766 0.15181
```

The following SPlus function is called "llclassify". It is an implementation of Livingston and Lewis' (1995) method for estimating the accuracy of classifications based on test scores. We have made a slight modification whereby in estimating the joint distribution of true scores and observed scores, we integrate over the true score distribution rather than dividing it into bins. This modification leads to simpler code, faster output, and greater precision. The main output is a classification matrix, where a row denotes the true score category and a column represents the observed score category.

The function is given below. Comments are denoted by #. The function call is of the form

```
function(obstable, r, max, cut, min = 0, adjust = 1).


# The arguments of this function are as follows:
# First argument is the inputted data. Data should
# be a table of numeric scores (either scale or
# number correct). The first column of the table
# should be possible scores, and the second column
# should be the number of students who obtained each
# corresponding score in the first column. Second
# argument is inputted reliability coefficient. Third
# argument is the maximum possible score on the test.
# Fourth argument is vector of cut points. Cuts are
# the lower bound of the "higher" category. For
# instance, if Far Below Basic stops at a score of
# 26, and Below Basic starts at 27, then put in 27
# for that cut. Fifth argument is minimum possible
# score on the test (default=0). Sixth argument is
# whether to adjust the results to match the
# marginals of the observed score distribution;
# default (1) is to adjust them. Put in a value other
# than 1 if you do not want to adjust the values in
# this way.


{
# Adjust the scores in case min is not 0, and do
# appropriate calculations.
        observedy <- rep(obstable[,1], obstable[,2])
        observed <- observedy - min
        max <- max - min
        cut <- cut - min
        x <- round(observed)
        p <- (x)/(max)
        mup <- sum(p)/length(p)
        deviations <- (p - mup)
        squaredeviations <- deviations^2
        sigma2p <- sum(squaredeviations)
        /(length(p))

# Find L&L's effective test length, and adjust
# observed scores according to effective test
# length.

        effective <- (mup * (1 - mup) - r *
        sigma2p)/(sigma2p * (1 - r))
        n <- round(effective)

# Fit 4-parameter beta distribution to true scores,
# using method of moments.

        xprimenotrounded <- (n * (x))/(max)
        xprime <- round(xprimenotrounded)
        mprime1x <- sum(xprime)/length(xprime)
        mprime2x <- sum(xprime^2)/length(xprime)
        mprime3x <- sum(xprime^3)/length(xprime)
        mprime4x <- sum(xprime^4)/length(xprime)
        mprime1p <- mprime1x/n
        mprime2p <- (mprime2x - mprime1x)/
        (n * (n - 1))
        mprime3p <- (mprime3x - 3 * mprime2x +
        2 * mprime1x)/(n * (n - 1) * (n - 2))
        mprime4p <- (mprime4x - 6 * mprime3x +
        11 * mprime2x - 6 * mprime1x)/
        (n * (n - 1) * (n - 2) * (n - 3))
        m2p <- (mprime2p - mprime1p^2)
        m3p <- (mprime3p - 3 * mprime1p * mprime2p
        + 2 * mprime1p^3)
        m4p <- (mprime4p - 4 * mprime1p * mprime3p
        + 6 * mprime1p^2 * mprime2p - 3 *
        mprime1p^4)
```

```
        k <- (16 * m2p^3)/m3p^2
        l <- (m4p * m2p)/m3p^2
        phi <- (3 * (k - 16 * (l - 1)))/
        (16 * (l - 1) - 8 - 3 * k)
        sgnm3p <- m3p/abs(m3p)

# Revert to 3-parameter beta distribution with
# beta=1 if 4-parameter method of moments does not
# give an appropriate solution.

        if(as.double(k * (phi + 1) + (phi + 2)^2)
        <= 0) {
        z <- (2 * (1 - mprime1p) * m2p)/m3p
        new <- m2p/(1 - mprime1p)^2
        betahat <- (z * (1 - new) - 2)/(1 - new +
        2 * new * z)
        alphahat <- (new * betahat * (1 + betahat))
        /(1 - new * betahat)
        ahat <- ((alphahat + betahat) * mprime1p -
        alphahat)/betahat
        bhat <- 1
        }

# Fit the 4-parameter beta distribution if method of
# moments gives an appropriate solution.  alphahat,
# betahat, ahat, and bhat are estimates of the 4
# respective parameters:  alpha, beta, a, and b.
# alpha and beta are the usual parameters of the
# 2-parameter beta distribution.  a is the lowest
# value that the true score can take, while b is the
# highest such value.

        else {
        theta <- (sgnm3p * phi * (phi + 2))/
        sqrt(k * (phi + 1) + (phi + 2)^2)
        alphahat <- (phi - theta)/2
        betahat <- (phi + theta)/2
        ahat <- (mprime1p - sqrt((m2p * alphahat
        * (alphahat + betahat + 1))/betahat))
        bhat <- (mprime1p + sqrt((m2p * betahat *
        (alphahat + betahat + 1))/alphahat))
        }
        if((ahat < 0) || (alphahat < 0) ||
        (betahat < 0)) {
        rrr <- m2p/mprime1p^2
        w <- (2 * m2p^2)/(mprime1p * m3p)
        alphahat <- (w * (rrr - 1) - 2)/(1 - rrr -
        2 * rrr * w)
        betahat <- (rrr * alphahat * (alphahat + 1))
        /(1 - rrr * alphahat)
        bhat <- ((alphahat + betahat) * mprime1p)
        /alphahat
        ahat <- 0
        }

# Fit a simple beta model if the method of moments
# continues not to give an appropriate solution.

        if((bhat > 1) || (alphahat < 0) || (betahat
        < 0)) {
        z <- (2 * (1 - mprime1p) * m2p)/m3p
        new <- m2p/(1 - mprime1p)^2
        betahat <- (z * (1 - new) - 2)/(1 - new +
        2 * new * z)
        alphahat <- (new * betahat * (1 + betahat))
        /(1 - new * betahat)
        ahat <- ((alphahat + betahat) * mprime1p -
        alphahat)/betahat
        bhat <- 1
        }

# Figure out which values inside [ahat, bhat] belong
# in which true score category.

        cutinp <- (cut - 0.5)/max
        cutinp2 <- c(ahat, cutinp, bhat)

# Figure out which observed scores will map into
# which observed classification.  1e-07 term
# necessary because of unexpected results from
# SPlus' "floor" function.

        cut2 <- floor(((cut - 0.5) * n)/max - 1e-07)
        cut3 <- c(0, cut2, n)

# Figure out each entry in confusion matrix, which
# gives the joint distribution of true score and
# observed score.  This program integrates over true
```

```
# score, rather than L&L's suggestion of dividing
# into bins.  This modification is both more exact
# and much faster.  Note that the additional
# function "ffunc" is required to run this program.
# ffunc is added at the end of this text.

        finalmatrix <- matrix(0, ncol = length(cut)
        + 1, nrow = length(cut) + 1)
        for(i in 1:nrow(finalmatrix)) {
        for(j in 1:nrow(finalmatrix)) {
        finalmatrix[i, j] <- integrate(ffunc,
        cutinp2[i], cutinp2[i + 1], ahat1 = ahat,
        bhat1 = bhat, alphahat1 = alphahat, betahat1
        = betahat, lowbinom = cut3[j], highbinom =
        cut3[j + 1], n1 = n)$integral
        }
        }

# If adjust is set to 1, adjust the entries to match
# the observed score distribution.

        if(adjust == 1) {
        proportions <- c(0, length(cut) + 1)
        xlowest <- x[x < cut[1]]
        proportions[1] <- length(xlowest)/length(x)
        if(length(cut) > 1) {
        for(i in 2:length(cut)) {
        temp <- x[x < cut[i]]
        proportions[i] <- (length(temp)/length(x)
        - sum(proportions[1:(i - 1)]))
        }
        }
        proportions[length(cut) + 1] <- 1 -
        sum(proportions[1:length(cut)])
        for(i in 1:ncol(finalmatrix)) {
        finalmatrix[, i] <- (finalmatrix[, i] *
        proportions[i])/sum(finalmatrix[, i])
        }
        }

# Return the parameters of the fitted 4-parameter
# beta distribution, the effective test length, and
# the confusion matrix.

        list(alphahat = alphahat, betahat = betahat,
        ahat = ahat, bhat = bhat, n = n,
        finalmatrix = finalmatrix)
}



# This short function "ffunc" is needed to perform
# the integral used in the above function.  All
# arguments are defined within the function that
# calls it.

> ffunc
function(x, ahat1, bhat1, alphahat1, betahat1,
lowbinom, highbinom, n1)
{
        (dbeta(((x - ahat1))/(bhat1 - ahat1),
        alphahat1, betahat1) * (pbinom(
        highbinom, n1, x) - pbinom(lowbinom, n1, x)))
        /(bhat1 - ahat1)
}

# The following is an example of the function's
# output.  It is based on 2003 raw score data of
# grade 4 math students.  $alphahat is the
# estimated value of alpha in the 4-parameter beta
# distribution.  $betahat is the estimated value of
# beta in this 4-parameter beta distribution.
# $ahat and $bhat are the estimated lower and upper
# bounds of this true score distribution.  $n is
# the so-called "effective test length" in the
# Livingston-Lewis paper.  $finalmatrix is the final
# confusion matrix outputted by the function.  Its
# entries have been rounded off for ease of viewing.

$alphahat:
[1] 1.377472

$betahat:
[1] 0.783828

$ahat:
[1] 0.2031712
```

```
$bhat:
[1] 0.9505744

$n:
[1] 68

$finalmatrix:

        [,1]    [,2]    [,3]    [,4]    [,5]
[1,] 0.04458 0.01162 0.00000 0.00000 0.00000
[2,] 0.01921 0.16683 0.02461 0.00001 0.00000
[3,] 0.00000 0.03097 0.21290 0.03775 0.00004
[4,] 0.00000 0.00000 0.03094 0.19988 0.03115
[5,] 0.00000 0.00000 0.00001 0.03766 0.15181
```