Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Supplementary Material for "FBB: A Fast Bayesian Bound tool to calibrate RNA-seq aligners"

Irene Rodriguez-Lujan[1,2], Jeff Hasty[1,3,4], Ramón Huerta[1,*]

[1] BioCircuits Institute, University of California, San Diego, La Jolla, CA 92093-0328, USA
[2] Machine Learning Group, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain
[3] Department of Bioengineering, University of California, San Diego, La Jolla, CA 92093, USA
[4] Molecular Biology Section, Division of Biological Science, University of California, San Diego, La Jolla, CA 92093, USA.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

1 Empirical estimate of paired-end read distances

Instead of estimating $P(d)$ in several iterations, we start with a uniform distribution between 0 and $3,000$ base pairs. Then, we count the number of events in an array $N(d)$. Every time a new paired-end read is a proper alignment; that is, there exists a solution to $\arg\max_{i,d}[P(i,i+d|\mathbf{r},\mathbf{r}')-\theta]_+$ with $P(i,i+d|\mathbf{r},\mathbf{r}')$ estimated as in Equation (16), then $N(d)$ is updated by increasing its value by 1. This is done to accelerate performance. It is not ideal because the distribution is less accurate in the initial reads than later ones, but we avoid running the algorithm twice. An example of the process is shown in Fig. 1, which corresponds to one of the data sets used in the Results section. The estimation of $P(d)$ approximates the stationary distribution after 100,000 paired-end reads in this case.

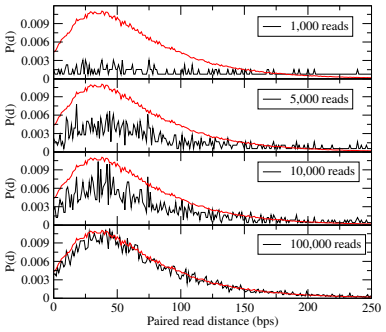


Fig. 1: Convergence of the estimation of the probability distribution of paired-end read distances according to the algorithm based on Theorem 2. The red line represent $P(d)$ after 16 million reads have been processed using the chromosome VII of the *Saccharomyces cerevisiae*. After 100,000 paired-end reads the distribution gets fairly close to the final distribution. d is calculated between the starting locations of the aligned paired-end reads.

2 Aligners Parametrization

These are the command line options used in the algorithms:

1. **Bowtie** was run with the following options: `-best -y -k 1 -I 0 -X 3100 -v`.
   - `-X 3100`: we limit the maximum distance between paired-end reads to 3100 bps in order to evaluate false paired alignments.
   - `-v 0,1,2,3`: we explore Bowtie outputs as function of 0,1,2 or 3 mismatches.
   - `-best -y -k 1`: best-hit reporting.
2. **SHRiMP2**: is an improved version of SHRiMP1 in (Rumble *et al.*, 2009). SHRiMP2 contains significant variations respect to the rest of aligners because it uses multiple seeds to filter the read before applying the Smith-Waterman algorithm. SHRiMP2 was run with the following options: `-I0,3100 -N  1 -max-alignments 1 -o 1 -strata -p opp-in -single-best-mapping -qv-offset 33 -min-avg-qv 0 -E -Q -h -m .`
   - `-I0,3100`: to limit the maximum and minimum distance between paired-end reads to have equivalent comparisons across all the aligners.
   - `-h`: the hit threshold of the Smith-Waterman algorithm. It varies from `-h90%` to `-h100%`.

- `-m`: Smith-Waterman match score varied from 1 to 10.
- `-max-alignments 1 -o 1 -strata -p opp-in -single-best-mapping`: best-hit reporting.

3. **Soap 2** is another fast implementation (Li *et al.*, 2009) that is very similar to Bowtie because it is a seed-and-extend algorithm. The conversion tool from Soap to SAM leaves the match paired flag unchecked. So we have to assume that the paired reads are called actual pairs by the algorithm. Otherwise there are no paired reads in the output SAM file. Soap was run with the following parameters: `-m 0 -x 3100 -M -r 1`

- `-x 3100 -m 0`: to limit the distance between paired-end reads between 100 and 1300 bps.
- `-M` : which is almost equivalent to the Bowtie `-v` and indeed generates very similar results.
- `-r 1` : best-hit reporting.

4. **Novocraft.com** also has a very fast commercial alignment implementation called `Novoalign`. Novoalign was run with the following options: `-R 0 -r A 1 -p -1 -i +- 0-3100 -t`

- `-i +- 0-3100`: to limit the distance between paired-end reads between 0 and 3100 bps.
- `-p -1`: to disable the filters based on quality scores. the `-p` option states: "If there are n or more bases with phred quality below t then the read is flagged as polyclonal and will not be aligned".
- `-t`: it indicates the maximum alignment score. We used values from 0 to 200. Novoalign allows to smoothly regulate performance by this command line option.
- `-R 0 -r A 1`: best-hit reporting.

5. **Bowtie 2** is the next evolution of Bowtie. It is slightly slower than Bowtie but it produces a higher percentage of alignments and slightly higher errors. Bowtie 2 was used with the following parameters: `-no-unal -no-mixed -t -I 0 -X 3100 -score-min C,`

- `-X 3100`: we limit the maximum distance between paired-end reads to 3100 bps in order to evaluate false paired alignments.
- `-score-min C,x` : where `x` takes values from 0 to 200. Bowtie 2 allows to smoothly regulate performance by this command line option.

6. **Segemehl** is a robust aligner (Hoffmann *et al.*, 2009) that scored very high in the study (Caboche *et al.*, 2014) using simulated data. The command options used are: `-r 1 -t 1 -I 3100 -A`

- `-A`: is the accuracy that loops from 1 to 100.
- `-I 3100`: is the maximum size of the inserts is set to 3100.
- `-r 1`: best-hit reporting.

7. **BWA-MEM** does not have a straightforward manner to control for the minimum and maximum distance between paired read ends, and the reporting policy. The command options used are `-A -B -O` .

- `-A` The score for a sequence match was iterated from 1 to 100.
- `-B` The penalty for a mismatch ranged from 1 to 100.
- `-O` The gap open penalty ranged from 1 to 20.

8. **FBBA** is the algorithm based on the Fast Bayesian Bound. It was run with the `-t` option that regulates the minimum Bayesian bound to accept a paired read. It was run from -25 to -300.

Other methods like YADA might perform well according to (Břinda *et al.*, 2016). However we lack the same control on the parameters as the methods shown above.

3 Simulated data

ART simulator for next-sequencing reads (Huang *et al.*, 2012) takes a FASTA genome reference sequence for its input, and then simulates NGS reads by emulating the sequencing process with built-in, technology-specific read error models and base quality scores profiles obtained empirically from large sequencing datasets. The ART simulator allows specifying the sequencing system of the built-in profile to be used for simulation. ART supports Roche's 454, Illumina's Solexa, and Applied Biosystems' SOLiD platforms. Simulated reads of *Escherichia coli str. K-12 substr. MG1655* transcriptome with length 100 were generated by using the Illumina's HiSeq 2000 sequencing system with a 20-fold read coverage. ART generates unstranded simulated reads and, therefore, only the FBB estimate for false positive reads (Definitions 1 and 1) is considered as source of error. Additionally, ART supports the three types of sequencing errors: mismatches, insertions, and deletions.The amount of mismatches in the simulations can be easily controlled by means of a parameter (*mismatch shift*) that shifts the quality scores. The larger the parameter, the lower number of substitutions. The relative presence of insertions and deletions is regulated by the probability of success of a binomial distribution.

To analyze the performance of FBB under different scenarios, the mismatches parameter took values in the set $\{-6, -4, -2, 0, 2, 4, 6\}$, while two different indel probabilities was considered: the default value based on empirical data, and 0.01. The mismatches parameter equals zero means that the distribution of quality scores is the one provided by ART and based on empirical data. The average number of mismatches per read is 3.32, 2.24, 1.52, 0.96, 0.61, 0.38, and 0.24 for *mismatch shift* parameter equals -6, -4, -2, 0, 2, 4, and 6, respectively. The default indel probability leads to a mean number of $1.7 \cdot 10^{-4}$ deletions per read, and $7.6 \cdot 10^{-5}$ insertions per read; while an indel probability of 0.01 produces a mean number of 0.47 deletions per read, and 0.47 insertions per read.

The mean and standard deviation of size of RNA fragments for paired-end simulations were set to 450 and 100, respectively. These values were obtained from the *E. Coli* real data. This experimental setup corresponds to the following command line (ART-GreatSmokyMountains):

```
art -ss HS20 -i <FASTA> -o <output directory>
-l 100 -f 10 -sam -ef -qs <mismatch shift>
-qs2 <mismatch shift>
-ir <insertion probability>
-dr <deletion probability>
-ir2 <insertion probability>
-dr2 <deletion probability>
-p -m 450 -s 100 -qL 0 -qU 40
```

Simulations provide us with a ground truth to show that FBB indeed helps in identifying incorrect alignments, and to determine whether the F^* score estimate is accurate. Table 1 shows the performance of the FBBA aligner based on the maximization of the FBB score for different simulation scenarios together with the rate of true positive, false positive, and false negative alignments. Note that when evaluating simulations, there cannot be true negative cases as the simulator only generates "positive" reads, so we chose the largest value of the alignment threshold $\theta = 300$ to align as much reads as possible. Results for mismatch parameter ≥ 0 are considered to avoid simulated reads to be significantly different from the ground truth. Results for indel probability equals 0.01 has been included in order to show the performance of the FBB score with indels; however, it should be noted that the presence of indels in RNA-seq data is highly unlikely (Minoche *et al.*, 2011). According to the results in Table 1, the FBBA algorithm provides a F1-score very close to 1, which means that maximizing the FBB score is an effective strategy that leads to identify the vast majority of correct alignments with a very low false positive rates.

Table 1. Average FBB score ($\langle FBB \rangle$), number of true True Positive (TP), False Positive (FP), and False Negative (FN) alignments and true F1-score for the FBBA algorithm with threshold parameter $\theta = 300$ tested on simulated reads from Escherichia coli str. K-12 substr. MG1655. Results for simulations with different densities of mismatches (mismatch shift parameter) and indel probabilities (Indel prob.) are shown. The most realistic scenario corresponds to Indel prob. and mismatches shift equals zero (in bold).

Indel prob.	Mismatch shift	$\langle FBB \rangle$	%TP	%FP	%FN	F1-score
0	**0**	**-56.70**	**99.41**	**0.58**	**0.58**	**0.9942**
	2	-43.85	99.64	0.36	0.36	0.9964
	4	-34.71	99.77	0.23	0.23	0.9977
	6	-28.60	99.86	0.14	0.14	0.9986
1	0	-77.71	96.88	3.11	3.12	0.9689
	2	-70.63	96.94	3.05	3.06	0.9694
	4	-65.25	97.03	2.97	2.97	0.9703
	6	-61.46	97.08	2.92	2.92	0.9708

Once the effectiveness of the FBB score to find correct alignments has been analyzed, we can make use of the known ground truth for alignments to determine whether the F^* score estimate is accurate. The relationship between the real F1-score (*F-score*) and the F1-score proxy (F^*) is presented in Figure 2, which shows the scatter plot between *F1-score* and F^* for different aligners and parameter settings. These results correspond to simulations with the two indel probabilities considered (default and 0.01), and three different concentrations of mismatches: $\{-2, 0, 2\}$. The bisector in black represents the case in which *F1-score* and F^* are identical. In all cases, points are mainly clustered around the bisector and, thus, Pearson's correlations between *F1-score* and F^* are very close to 1, even when there is a high presence of mismatches (lowest values of *mismatch shift*) or indels (*indel probability* 0.01). As expected, it can be seen that F^* converges to the real F1-score as the proportion of mismatches decreases (*mismatches shift* increases), and points are always below the bisector as F^* is a lower bound of the true F1-score. In fact, F^* seems to underestimate Segemehl's true F-score, which partially explains the results obtained for real data. Overall, Figure 2 shows that F^* is a reasonable proxy for the F1-score.

Finally, Figure 3 shows the percentage of true positives alignments, the percentage of false positive alignments, the true F1-score, and the estimation of the F1-score as a function of $\langle FBB \rangle$ for the two different indel probabilities considered and the empirical quality scores provided by ART simulator. True positives, false negatives, and F1-score are computed based on the known ground truth from the simulations. In the most realistic situation (Fig. 3a), results are similar to those obtained for real data. Most aligners perform similarly for a given value of $\langle FBB \rangle$, except for BWA MEM, which has the lowest alignment rates and the largest false positive percentages. In simulated data, Segemehl is slightly behind most aligners in terms of the true F1-score, though the difference is smaller than in the case of the F1-score proxy F^*. As discussed in Figure 2, F^* underestimates the true F1-score for Segemehl. On the other hand, for larger level of indels, Fig. 3b shows that FBBA and Segemehl outperform other methods in terms of the F-score. These results are also consistent with those presented in (Holtgrewe *et al.*, 2011) for simulated data with indels, where three of these aligners (Bowtie, Soap2, and Shrimp 2) were evaluated to conclude that Shrimp2 yields better performance than Bowtie and Soap2 in this case as Bowtie and Soap2 do not support indels.

3.1 Comparison to Cadbure

In this section, we contrast FBB results with the output provided by Cadbure (Kumar *et al.*, 2015), one of the state-of-the-art approaches to compare aligners that does not need simulated data either. Though not needed, we carried out this comparison on simulated data used in the previous section in order to have a ground truth against which we can fairly compare both algorithms. Cadbure makes pairwise comparisons between aligners and parameter configurations, so if we want to compare all the aligners and parameter configurations here considered (Section 2), the computational cost of taking all the possible pairs of aligners and parameters is computationally very expensive. If we want to compare N aligners and p different configurations in each of them, Cadbure needs to perform $\binom{N}{2}p^2$ comparisons; that is, the computational cost is $O(N^2p^2)$, which is unfeasible for not so large values of N and p. In contrast, FBB provides a proxy for the F1-score for each aligner and each parameter configuration, then it needs $O(Np)$ comparisons, which is a linear growth with respect to N and p instead of quadratic. As it is shown below, Cadbure and FBB provide similar results, but the computational cost of FBB is significantly lower.

Using the simulated data of *Escherichia coli str. K-12 substr. MG1655* genome, we compare Cadbure and FBB by taking as reference the ground truth from simulations. This way, we can determine how close is Cadbure's F1-score to FBB's F1-score proxy F^*, and we can also compare these results with those obtained in Figure 2. Moreover, this experimental setup allows us to reduce Cadbure's computational cost since we only compare each aligner and parameter setting against simulations' ground truth instead of making all possible pairwise comparisons. Cadbure software does not report a value for the F1-score, but it provide estimates for the number of True Positives (TP), False Positives (FP), and True Negatives (TN) alignments. Since F1-score is a function of TP, FP and False Negatives (FN), we obtained the number of FN for an aligner X as the number of reads that the other aligner Y maps uniquely (TP), but aligner X does not map either uniquely or non-uniquely.

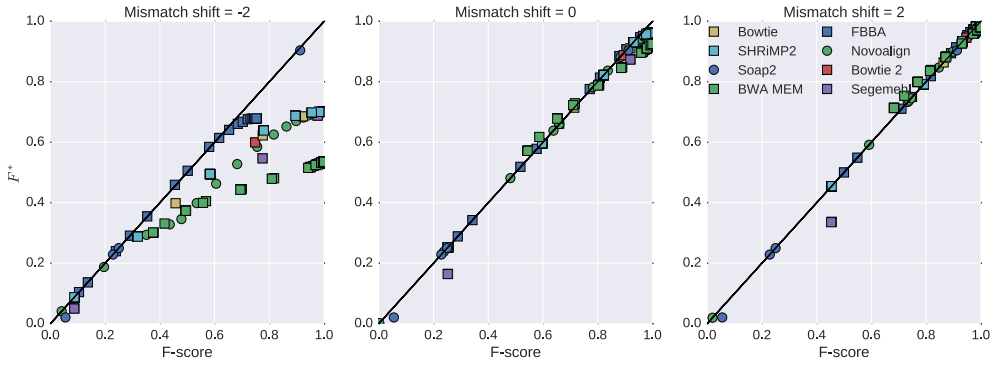
The scatter plot between Cadbure's F1-score (F_C) and FBB's proxy (F^*) is presented in Figure 4. In this case, the bisector represents the case in which Cadbure and FBB estimates match. It can be seen that both algorithms provide very similar results in general, when there is not a high density of mismatches. Though they significantly differ in their estimates for BWA MEM, their outputs for Bowtie, and Segemehl are slightly different in some cases, and they yield dissimilar results for large mismatches rates (*mismatches shift* equals -2), Pearson's correlations between F_C and F^* are still larger than 0.68 for all the aligners and all mismatches and indels rates. If BWA MEM is not considered in the computation, the lowest Pearson's correlation for all scenarios is above 0.95. While differences for BWA MEM and Segemehl with respect to other mappers on real date may be partially explained by the FBB's F-score estimates (F^*), which are lower than the true F-score as shown in Fig. 2, dissimilarities for Bowtie have to be due to Cadbure's misestimates of the F-score as F^* is highly accurate for Bowtie in most cases.

4 Computational times

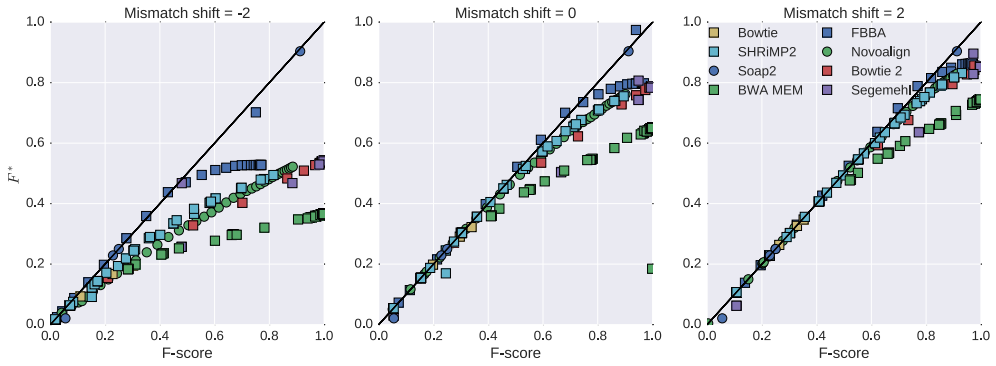
Table 2 shows the average computational times to compute the FBB score from a SAM input file (FBB), and to align reads by means of the FBBA strategy (FBBA). FBB computational times are obtained averaging over all the aligners considered and all the configuration parameters (see Section 2). FBBA computational times correspond to averaging over all the FBBA parameter settings. It should be noted that FBBA evaluates as many reads as there are in FASTQ files, while FBB evaluates the valid alignments reported by each mapper.

References

Břinda,K., Boeva,V. and Kucherov,G. (2016) RNF: a general framework to evaluate NGS read mappers. *Bioinformatics,* **32** (1), 136–139.



(a) Default indel probability



(b) Indel probability 0.01

Fig. 2: Scatter plot of the real F1-score (*F-score*) versus the proxy F1-score (F^*) for different aligners (Bowtie, SHRiMP2, Soap 2, BWA MEM, FBBA, Novoalign, Bowtie 2, and Segemehl) on simulated reads from *Escherichia coli str. K-12 substr. MG1655*. Results for simulations with different densities of mismatches (*mismatch shift* parameter) and indel probabilities are presented. The most realistic scenario corresponds to Fig. 2a with *mismatches shift* equals zero and default indel probability.

Table 2. Average computational time per read in milliseconds of FBB and FBBA for the three real RNA-seq datasets on Intel(R) Xeon(R) CPU E5-2680 0 @ 2.70GHz.

Dataset	FBB	FBBA
Saccharomyces	0.019322	0.018919
Mus musculus	2.175024	0.085874
E.Coli	0.017929	0.032418

Caboche,S., Audebert,C., Lemoine,Y. and Hot,D. (2014) Comparison of mapping algorithms used in high-throughput sequencing: application to Ion Torrent data. *BMC genomics,* **15** (1), 264.

Hoffmann,S., Otto,C., Kurtz,S., Sharma,C.M., Khaitovich,P., Vogel,J., Stadler,P.F. and Hackermüller,J. (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput Biol,* **5** (9), e1000502.

Holtgrewe,M., Emde,A.K., Weese,D. and Reinert,K. (2011) A novel and well-defined benchmarking method for second generation read mapping. *BMC bioinformatics,* **12** (1), 1.

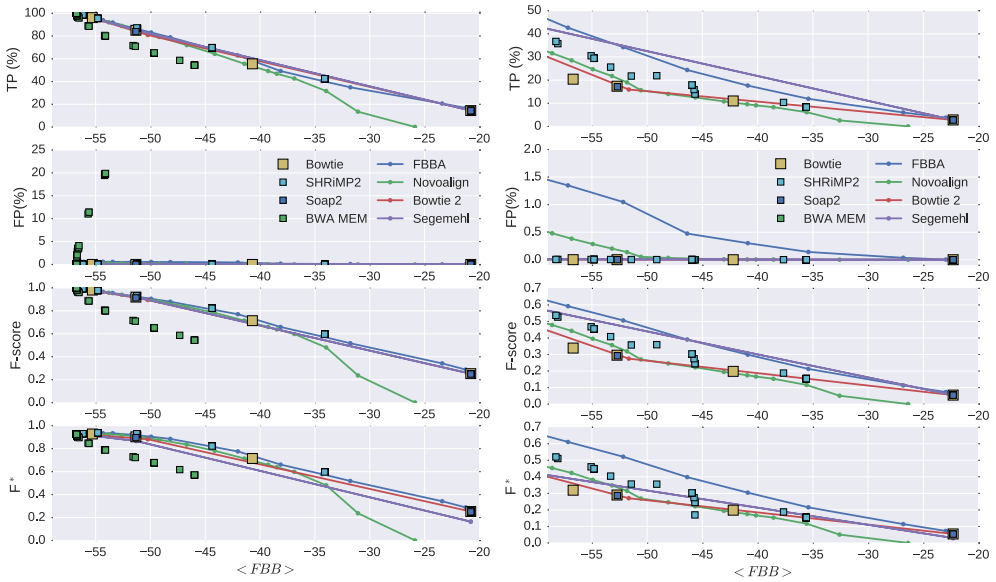
Huang,W., Li,L., Myers,J.R. and Marth,G.T. (2012) ART: a next-generation sequencing read simulator. *Bioinformatics,* **28** (4), 593–594.

Kumar,P.K.R., Hoang,T.V., Robinson,M.L., Tsonis,P.A. and Liang,C. (2015) CADBURE: A generic tool to evaluate the performance of spliced aligners on RNA-Seq data. *Scientific reports,* **5**.

Li,R., Yu,C., Li,Y., Lam,T.W., Yiu,S.M., Kristiansen,K. and Wang,J. (2009) SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics,* **25** (15), 1966–1967.

Minoche,A.E., Dohm,J.C. and Himmelbauer,H. (2011) Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. *Genome biology,* **12** (11), 1.

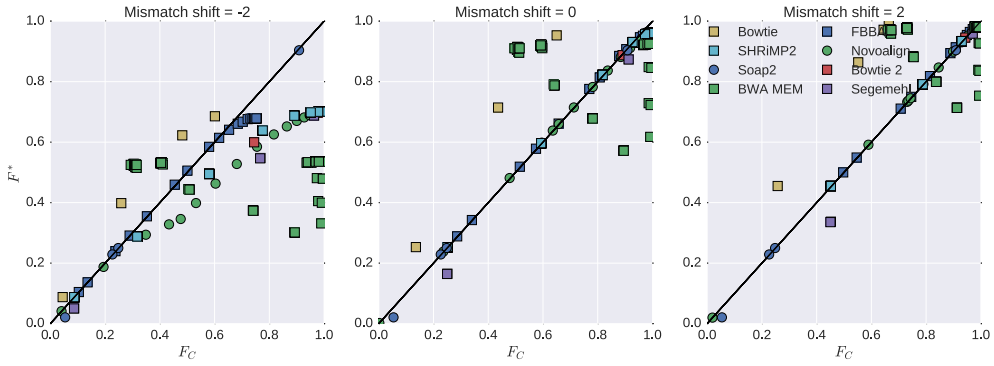
Rumble,S.M., Lacroute,P., Dalca,A.V., Fiume,M., Sidow,A. and Brudno,M. (2009) SHRiMP: accurate mapping of short color-space reads. *PLoS Comput Biol,* **5** (5), e1000386.



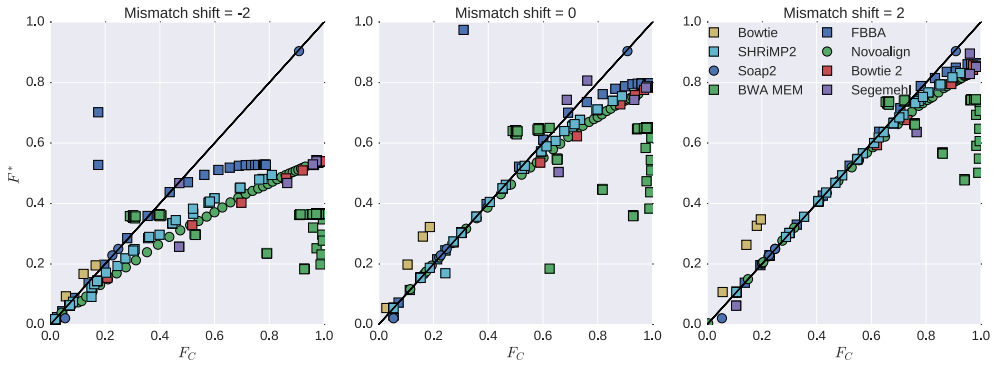
(a) Default indel probability

(b) Indel probability 0.01

Fig. 3: Alignment comparison of Bowtie, SHRiMP2, Soap 2, BWA MEM, FBBA, Novoalign, Bowtie 2, and Segemehl on simulated data of *Escherichia coli str. K-12 substr. MG1655* as a function of the average FBB values, $\langle FBB \rangle$. Results correspond to simulations with unaltered q-scores distribution (*mismatches shift* equals 0), and either (a) default indel probability or (b) indel probability equals 0.01. (first) Percentage of True Positive (TP) alignments based on ground truth. (second) percentage of False Positive alignments (FP) based on ground truth, (third) true F1-score, and (fourth) F^*, a proxy of the F1-score.



(a) Default indel probability



(b) Indel probability 0.01

Fig. 4: Scatter plot of Cadbure's F1-score (F_C) versus the proxy F1-score (F^*) for different aligners (Bowtie, SHRiMP2, Soap 2, BWA MEM, FBBA, Novoalign, Bowtie 2, and Segemehl) on simulated reads from *Escherichia coli str. K-12 substr. MG1655* are shown. Results for simulations with different densities of mismatches (*mismatch shift* parameter) and indel probabilities. The most realistic scenario corresponds to Fig. 4a with *mismatches shift* equals zero and default indel probability.