

31 **ABSTRACT** Microsatellite (SSR) is one of the most popular markers for applied genetic research, but
32 generally the current methods to develop SSRs are relatively time-consuming and expensive. Although
33 high-throughput sequencing (HTS) approach has become a practical and relatively inexpensive option
34 so far, only a small percentage of SSR markers turn out to be polymorphic. Here, we designed a new
35 method to enrich polymorphic SSRs through the comparative transcriptome analysis. This program
36 contains five main steps: 1) transcriptome data downloading or RNA-seq; 2) sequence assembly; 3)
37 SSR mining and enrichment of sequences containing SSRs; 4) sequence alignment; 5) enrichment of
38 sequences containing polymorphic SSRs. A validation experiment was performed and the results
39 showed almost all markers (> 90%) that were indicated as putatively polymorphic by this method were
40 indeed polymorphic. The frequency of polymorphic SSRs was significantly higher ($P < 0.05$) but the
41 cost and running time were much lower than those of traditional and HTS approaches. The method has
42 a practical value for polymorphic SSRs development and might be widely used for genetic analyses in
43 any species.

44

45

46 INTRODUCTION

47 Microsatellites (SSRs) have been emerged as one of the most popular markers for a wide range of
48 applications in population genetics, conservation biology and marker-assisted selection (Abdelkrim et
49 al., 2009; Luo et al., 2012). Classically, microsatellite marker development requires: the construction of
50 a genomic library enriched for repeated motifs; isolation and sequencing of microsatellite containing
51 clones; primer design; optimization of PCR amplification for each primer pair; and a test of
52 polymorphism on a few unrelated individuals. Most of these steps are either expensive,
53 time-consuming, or both. With the wide application of high-throughput sequencing (HTS) technology,
54 especially the whole transcriptome sequencing, development SSRs by HTS has become a practicable
55 alternative for many species in recent years (Wu et al., 2014). It has greatly reduced the running time
56 and cost requirement for SSR development. However, the frequency of polymorphic SSR markers
57 developed by this method is much low in some species, which means that most of the loci cannot be
58 effectively applied in genetic analysis (Iorizzo et al., 2011; Luo et al., 2012). According to our best
59 knowledge, there is no records addressed the low frequency of polymorphic SSRs. Here we provided a
60 new method for development of polymorphic SSRs through comparative transcriptome analysis.

61

62 RESULTS AND DISCUSSION

63 Three, two and four transcriptomes of rice, grass carp and lined seahorse respectively were used and a
64 total of 299, 206 and 956 putatively polymorphic SSRs were obtained by this method, respectively
65 (Table 1; Table S1). Twenty, thirty and sixty loci were randomly selected for primer design, and 19
66 (95.00%), 26 (92.86%) and 50 (90.91%) loci showed polymorphic in rice, grass carp and lined
67 seahorse, respectively (Table 1; Table S2). One-way ANOVA showed the frequency of polymorphic
68 SSRs identified by this method was significantly ($P < 0.05$) higher than that of traditional approach and
69 HTS (Fig. 2). In addition, we recently developed polymorphic SSR markers for lined seahorse by HTS
70 approach, and the ratio of polymorphic SSRs was 17.93% (Arias et al., 2016). While using the same
71 transcriptomes to develop SSRs by this method, the ratio was raised to 90.91%.

72 This method, which is based on the idea of enriching homologous sequences containing the same
73 SSR with a different number of repeats, could identify polymorphic SSRs directly through comparative
74 transcriptome analysis. Compared with traditional methods and HTS, this method have eliminated the
75 most intensive wet lab steps, and the time and cost for primer synthesis and experimental validation by

76 identifying and separating the many “monomorphic” SSRs from the minority polymorphic ones,
77 cutting the running time and cost by half or more (Tang et al., 2008; Iorizzo et al., 2011). The fact that
78 almost all tested SSRs predicted to be polymorphic were indeed validated as polymorphic demonstrates
79 that it is an efficient and reliable method to develop polymorphic SSR markers. The method will play
80 an important role in developing polymorphic SSR markers, providing a better service for the selective
81 breeding and genetic studies.

82

83 **MATERIALS AND METHODS**

84 **Materials**

85 Each ten specimens of rice (*Oryza sativa*), grass carp (*Ctenopharyngodon idella*) and lined seahorse
86 (*Hippocampus erectus*) were used to investigate the ratio of polymorphic SSRs developed by our
87 method. The leaves of rice and the dorsal fin of grass carp and seahorse were used for DNA extraction.

88 **Architectural structure**

89 The pipeline of this method consists of five steps (Fig. 1): 1) transcriptome data downloading or
90 RNA-seq; 2) sequence assembly; 3) SSR mining and enrichment of SSR containing sequences; 4)
91 sequence alignment; 5) enrichment containing polymorphic microsatellite sequences.

92 **Transcriptome data gaining**

93 The precondition of this method is to detect polymorphic SSRs in two or more transcriptomes from
94 different samples. Three and two transcriptomes of rice and grass carp were downloaded from NCBI
95 (Table 1). Four transcriptomes of seahorse were sequenced by us.

96 **De novo assembly**

97 The raw reads were trimmed and quality controlled by SeqPrep (<https://github.com/jstjohn/SeqPrep>).
98 Then clean data was used to perform RNA *de novo* assembly with Trinity using default parameters.

99 **SSR mining and enrichment of SSR containing sequences**

100 We took rice as an example to enrich polymorphic SSRs. The three transcriptomes assembled were
101 renamed “T1”, “T2” and “T3”, respectively.

102 MicroSAteLLite identification tool (MISA; <http://pgrc.ipk-gatersleben.de/misa>) was employed for
103 SSR mining from different transcriptomes with the following settings (SSR motifs and number of
104 repeats): dimer-6, trimer-5, tetramer-5, pentamer-5 and hexamer-5. In order to reduce the rate of false

105 positives, a Python code was written to rule out the sequences which only contain mononucleotide
106 repeats, compound SSRs, or end with SSRs. The command line is written as follows:

```
107     from Bio import SeqIO
108     import os
109     samples=['T1','T3','T4']
110     for sample in samples:
111         ids=[]
112         faD=SeqIO.to_dict(SeqIO.parse(open(sample+'.fa'),'fasta'))
113         for la in open(sample+'.ssr'):
114             if 'ID' not in la:
115                 aL=la.split('\t')
116                 ln=len(faD[aL[0]].seq)
117                 if aL[2]!='p1' and aL[5]!=1 and aL[6]!=ln:
118                     ids.append(aL[0])
119     comp_ids=[]
120     for lb in open(sample+'.ssr'):
121         bL=lb.strip().split('\t')
122         if 'c' in bL[2]:
123             comp_ids.append(bL[0])
124     fas=SeqIO.parse(open(sample+'.fa'),'fasta')
125     for fa in fas:
126         if fa.id in ids and fa.id not in comp_ids:
127             open(sample+'.ssr;fa','a').write('>%s\n%s\n' % (fa.id,str(fa.seq)))
```

128
129 And then we renamed all the transcriptomes and all the sequences containing SSRs detected with
130 MISA software by adding different prefixes. Finally, we combined all the sequences containing SSRs
131 from different transcriptomes into a file. The command line is written as follows:

```
132     from Bio import SeqIO
133     samples=['T1','T3','T4']
134     for sample in samples:
```

```
135     fas=SeqIO.parse(open(sample+'.Trinity.fasta'),'fasta')
136     for fa in fas:
137         open(sample+'.fa','a').write('>%s.%s\n%s\n' % (sample,fa.id,str(fa.seq)))
138     for la in open(sample+'.Trinity.fasta.misa'):
139         if 'ID' not in la:
140             la=sample+'.'+la
141     open(sample+'.ssr','a').write(str(la))
```

142

143 **Alignment of containing SSR sequences**

144 Sequences containing SSRs were clustered using the default parameters of the CD-HIT tool at 90%
145 sequence identity level.

146 **Rename the SSR file**

147 We then edited a Python code that generated the reverse complement of the minus strand transcripts,
148 according to the strand information in the output of CD-HIT:

```
149     import re
150     from Bio import SeqIO
151     faD=SeqIO.to_dict(SeqIO.parse(open('all.ssr.fa'),'fasta'))
152     baseD={'A':'T','T':'A','G':'C','C':'G'}
153     samples=['T1','T3','T4']
154     for sample in samples:
155         for line in open(sample+'.ssr'):
156             lst=line.strip().split('\t')
157             if lst[0] in faD:
158                 if 'c' not in lst[2] and lst[2]!='p1' and 'ID' not in line and lst[-2]!='1' and
159 int(lst[-1])!=len(faD[lst[0]].seq) and int(lst[-1])!=len(faD[lst[0]].seq)-1 and
160 int(lst[-1])!=len(faD[lst[0]].seq)-2 and int(lst[-1])!=len(faD[lst[0]].seq)-3:
161                     ma=re.findall('\((.+)\)',lst[3])[0]
162                     maL=list(ma)
163                     rc=""
```

```
164         for base in maL:
165             if base in 'ACGT':
166                 rc += baseD[base]
167             rc = rc[::-1]
168             ss = [ma, rc]
169             ss.sort()
170             ss = '+'.join(ss)
171             newline = re.sub('\([ACGT]+\)', ss, line)
172             open(sample + '.ssr.reformed', 'a').write(str(newline))
173
```

174 And then, we edited a Python code that generated reverse complementary of minus strand transcripts,
175 according to the strand information in the output of CD-HIT:

```
176     from Bio import SeqIO
177     import re
178     sD = {}
179     for la in open('cd-hit.clstr'):
180         if 'at' in la:
181             id = re.findall('>(.)\.\.\.', la)[0]
182             strand = re.findall('[+-]+V', la)[0]
183             sD[id] = strand
184         if '*' in la:
185             id = re.findall('>(.)\.\.\.', la)[0]
186             sD[id] = '+'
187     fas = SeqIO.parse(open('all.ssr.fa'), 'fasta')
188     for fa in fas:
189         if fa.id in sD:
190             if sD[fa.id] == '+':
191                 open('plus.ssr.fa', 'a').write('>' + str(fa.id) + '\n' + str(fa.seq) + '\n')
192             if sD[fa.id] == '-':
193                 seq = fa.seq.reverse_complement()
```

```
194         open('plus.ssr:fa','a').write('>'+str(fa.id)+'\n'+str(seq)+'\n')
```

```
195
```

196 **Enrichment of sequences containing polymorphic SSRs**

197 A script was executed to enrich sequences with different repeats from all the sequences containing

198 SSRs:

```
199     import re
```

```
200     from collections import defaultdict
```

```
201     from Bio import SeqIO
```

```
202     import os
```

```
203     def getD(ssr):
```

```
204         s=[]
```

```
205         for la in open(ssr):
```

```
206             if 'ID' not in la:
```

```
207                 aL=la.strip().split('\t')
```

```
208                 ma=re.findall('^(.+)\d+',aL[3])
```

```
209                 s.append((aL[0],ma[0]))
```

```
210     d=defaultdict(set)
```

```
211     for k, v in s:
```

```
212         d[k].add(v)
```

```
213     return d
```

```
214     t1D=getD("T1.ssr.reformed")
```

```
215     t2D=getD("T3.ssr.reformed")
```

```
216     t3D=getD("T4.ssr.reformed")
```

```
217     allD={}
```

```
218     allD.update(t1D)
```

```
219     allD.update(t2D)
```

```
220     allD.update(t3D)
```

```
221     page=open('cd-hit.clstr').read()
```

```
222     clusters=re.findall('(.*?)>Cluster',page,re.S)
```

```
223     for cluster in clusters:
```

```
224     trans=re.findall('T\d,TR\d+\|c\d+_g\d+_i\d+',cluster,re.S)
225     if len(trans)>1:
226         tt=[]
227         ss=[]
228         for tran in trans:
229             if tran in allD:
230                 tt.append(str(tran)+'_'+str(list(allD[tran])))
231                 ss+=list(allD[tran])
232         if len(tt)>1:
233             ma=re.findall('\d+',str(ss))
234             ma=set(ma)
235             mas=re.findall('\((.+?)\)',str(set(ss)))
236             ssr=""
237             for mm in list(set(mas)):
238                 if mas.count(mm)>1:
239                     ssr=mm
240             if len(ma)>1 and len(mas)>len(set(mas)):
241                 ttt=[]
242                 for t in tt:
243                     if ssr in t:
244                         ttt.append(t)
245                 na=str(ttt).lstrip('[').rstrip(']').strip('>').replace("'", "'\t'")
246                 open('enrichment.SSRs','a').write(str(na)+'\n')
247         for ll in open('enrichment.SSRs'):
248             ma=re.findall('([a-zA-Z0-9]+\|c\d+_g\d+_i\d+)',ll)
249             ma=set(ma)
250             if len(ma)>1:
251                 open('enrichment.SSRs.txt','a').write(str(ll))
252         faD=SeqIO.to_dict(SeqIO.parse(open('plus.ssr.fa'),'fasta'))
253         n=0
```

```
254     for la in open('enrichment.SSRs.txt'):
255         aL=la.strip().split('\t')
256         for it in aL:
257             id=it.split(':')[0]
258             open('cluster'+str(n),'a').write('>'+str(id)+'\n'+str(faD[id].seq)+'\n')
259             os.system('muscle -msf -in cluster'+str(n)+' -out cluster'+str(n)+'muscle')
260             n+=1
261 os.system('mkdir muscle; mv cluster* muscle; rm enrichment.SSRs')
```

262

263 **Validation experiments**

264 Primers were designed using Primer premier 5 software. The PCR products were separated by capillary
265 gel electrophoresis using the ABI 3100 Genetic Analyser. The peak heights and fragment sizes were
266 analyzed using GeneMarker software.

267

268 **Data analysis**

269 Previous studies on SSR development in animals and plants, along with their frequency of polymorphic
270 SSR markers, were randomly downloaded from the internet (Table S3). Differences in mean value of
271 the frequency of polymorphic SSRs developed by three methods were analyzed using one-way analysis
272 of variance (ANOVA).

273

274 **Competing interests**

275 The authors declare no conflict of interest.

276

277 **Author contributions**

278 W.L. and Q.L. conceived and designed the experiments; H.Q., X.W. and Q.Z. performed the
279 experiments and analyzed data; W.L., and Q.L. prepared the manuscript. All authors read and approved
280 the final manuscript.

281

282 **Funding**

283 This study was funded by the National Natural Science Foundation of China (No. 41576145) and the
284 Guangdong Oceanic and Fisheries Science and Technology Foundation (No. A201501A12).

285

286 **Supplementary information**

287 **Table S1.** The putative polymorphic SSRs enriched by this method in rice (*O. sativa*), grass carp (*C.*
288 *idella*) and lined seahorse (*H. erectus*).

289 **Table S2.** The SSR loci validated by experiments and characteristics of the primers.

290 **Table S3.** The species and the frequency of polymorphic SSRs cited in this study.

291

292 **References**

293 **Abdelkrim, J., Robertson, B., Stanton, J. A. and Gemmell, N.** (2009). Fast, cost-effective
294 development of species-specific microsatellite markers by genomic sequencing. *Biotechniques* **46**,
295 185-192.

296 **Arias, M. C., Aulagnier, S., Baerwald, E. F., et al.** (2016). Microsatellite records for volume 8, issue
297 1. *Conserv. Genet. Resour.* **8**: 43-81.

298 **Iorizzo, M., Senalik, D. A., Grzebelus, D., Bowman, M., Cavagnaro, P. F., Matvienko, M., Ashrafi,**
299 **H., Van Deynze, A. and Simon, P. W.** (2011). *De novo* assembly and characterization of the
300 carrot transcriptome reveals novel genes, new markers, and genetic diversity. *BMC Genomics* **12**:
301 389.

302 **Luo, W., Nie, Z., Zhan, F., Wei, J., Wang, W. M. and Gao, Z. X.** (2012). Rapid development of
303 microsatellite markers for the endangered fish *Schizothorax biddulphi* (Günther) using next
304 generation sequencing and cross-species amplification. *Int. J. Mol. Sci.* **13**: 14946-14955.

305 **Tang, J., Baldwin, S. J., Jacobs, J. M., van der Linden, C. G., Voorrips, R. E., Leunissen, J. A. M.,**
306 **van Eck, H. and Ben, V.** (2008). Large-scale identification of polymorphic microsatellites using
307 an in silico approach. *BMC Bioinformatics* **9**: 374.

308 **Wu, T., Luo, S., Wang, R., Zhong, Y. J., Xu, X. M., Lin, Y. E., He, X. M., Sun, B. J. and Huang, H.**
309 **X.** (2014). The first Illumina-based de novo transcriptome sequencing and analysis of pumpkin
310 (*Cucurbita moschata* Duch.) and SSR marker development. *Mol. Breeding* **34**: 1437-1447.

311

312

313 **Table 1** Number of SSRs, polymorphic SSRs frequency of three verified species

	Species for verification		
	Rice	Grass carp	Lined seahorse
Source of the transcriptome	SRR1799209 SRR1974265 SRR2048540	SRR1618540 SRR1618542	SCSIO-CAS
Total number of SSRs in transcriptome	29,517	21,959	19,006
Number of putatively polymorphic SSRs enriched by this program	299	206	600
Number of primers designed for validation experiments	20	30	60
Number of primers amplified with clear product bands (%)	20 (100%)	28 (93.33%)	55 (91.7%)
Number of polymorphic SSRs (%)	19 (95.00%)	26 (92.86%)	50 (90.91%)

314

315

316

317 **Figure Legends :**

318 **Figure 1. Flowchart of polymorphic markers enrichment and development.**

319 **Figure 2. Comparison of polymorphic marker frequency developed by different methods.** A, B

320 and C represents SSRs developed by traditional methods, HTS approach and the method designed in

321 this study, respectively; * and ** represents significance at $P = 0.05$ and $P = 0.01$, respectively; #

322 represents SSR developed by HTS; † represents SSR developed by this method. The data of the

323 frequency of polymorphic SSRs used in the figure was cited from the published references (Table S1).

RNA



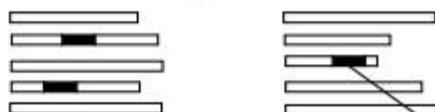
Step 1. Transcriptome data downloading or RNA-Seq

Transcriptome data are downloaded from the database; or total RNA is sequenced by next-generation sequencing technology.



Step 2. Sequence assembly

Raw reads are assembled using Trinity.



Step 3. SSR mining and enrichment of containing SSR sequences

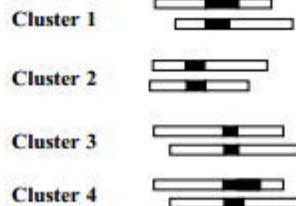
MISA is used to detect the reads containing microsatellite repeated motifs; The containing SSRs reads are enriched by this program.



SSR motif

Step 4. Sequence alignment

Sequences which contain SSRs are done sequence alignment by this program.



Step 5. Enrichment containing polymorphic microsatellite sequences

Sequences which contain SSRs with different repeat numbers are enriched.



The percentage of polymorphic SSRs (%)

