Computational and Structural Biotechnology Journal 19 (2021) 3069–3076

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

Modifying splice site usage with *ModCon*: Maintaining the *genetic code* while changing the underlying *mRNP code*

Johannes Ptok [a], Lisa Müller [a], Philipp Niklas Ostermann [a], Anastasia Ritchie [a], Alexander T. Dilthey [b,c,d], Stephan Theiss [a,*], Heiner Schaal [a,*]

[a] Institute of Virology, Medical Faculty, Heinrich Heine University Düsseldorf, D-40225 Düsseldorf, Germany
[b] Institute of Medical Statistics and Computational Biology, University of Cologne, Cologne, Germany
[c] Cologne Excellence Cluster on Cellular Stress Responses in Aging-Associated Diseases (CECAD), University of Cologne, Cologne, Germany
[d] Institute of Medical Microbiology and Hospital Hygiene, Heinrich Heine University Düsseldorf, Düsseldorf, Germany

ARTICLE INFO

Article history:
Received 16 February 2021
Received in revised form 14 May 2021
Accepted 20 May 2021
Available online 21 May 2021

Keywords:
pre-mRNA splicing
Splicing reporter
HEXplorer score
HBond score
Splice donor
Splicing regulatory proteins

ABSTRACT

Codon degeneracy of amino acid sequences permits an additional "mRNP code" layer underlying the genetic code that is related to RNA processing. In pre-mRNA splicing, splice site usage is determined by both intrinsic strength and sequence context providing RNA binding sites for splicing regulatory proteins. In this study, we systematically examined modification of splicing regulatory properties in the neighborhood of a GT site, i.e. potential splice site, without altering the encoded amino acids.

We quantified the splicing regulatory properties of the neighborhood around a potential splice site by its *Splice Site HEXplorer Weight* (SSHW) based on the HEXplorer score algorithm. To systematically modify GT site neighborhoods, either minimizing or maximizing their SSHW, we designed the novel stochastic optimization algorithm *ModCon* that applies a genetic algorithm with stochastic crossover, insertion and random mutation elements supplemented by a heuristic sliding window approach.

To assess the achievable range in SSHW in human splice donors without altering the encoded amino acids, we applied ModCon to a set of 1000 randomly selected Ensembl annotated human splice donor sites, achieving substantial and accurate changes in SSHW. Using ModCon optimization, we successfully switched splice donor usage in a splice site competition reporter containing coding sequences from FANCA, FANCB or BRCA2, while retaining their amino acid coding information.

The ModCon algorithm and its R package implementation can assist in reporter design by either introducing novel splice sites, silencing accidental, undesired splice sites, and by generally modifying the entire mRNP code while maintaining the genetic code.

1. Introduction

During splicing, introns of a newly synthetized pre-mRNA strand are mostly co-transcriptionally removed from the transcript, followed by the ligation of the remaining exonic sequence segments [1,2]. The intron excision cellular machinery is called spliceosome and recognizes canonical sequences with a GT dinucleotide at the upstream end of an intron, the 5′ splice site (5′ss) or splice donor (SD), and with an invariant AG at the downstream end of an intron, the 3′ splice site (3′ss) or splice acceptor (SA) [3,4].

Recognition of splice donor sites during spliceosome formation is accomplished through RNA duplex formation with 11 nucleotides of the free 5′end of the U1 snRNA [5], while splice acceptor sites are bound by U2 auxiliary factors (U2AF). A higher U1 snRNA complementarity is beneficial for 5′ splice site recognition and usage [6]. Several algorithms are available for scoring splice site strength: e.g. maximum entropy algorithms providing maxent scores for both 5′ss and 3′ss [7], and the HBond score reflecting 5′ss complementarity to U1 snRNA [5].

Abbreviations: A, adenine; eGFP, enhanced green fluorescent protein; F1, filial sequence 1; G, guanine; GA, genetic algorithm; hnRNP, heterogeneous nuclear ribonucleoproteins; HBS, HBond score; HZ_{EI}, HEXplorer score; nt, nucleotides; P1, parental sequence 1; pre-mRNA, precursor messenger RNA; SA, splice acceptor; SD, splice donor; snRNA, small nuclear RNA; SR proteins, serine- and arginine-rich proteins; SRP, splicing regulatory protein; SSHW, splice site HEXplorer weight; SW, sliding window; T, thymine.

* Corresponding authors.

https://doi.org/10.1016/j.csbj.2021.05.033

Beyond the proper splice site consensus sequences, splice site recognition has been shown to greatly depend on proximal binding of splicing regulatory proteins (SRPs) [8-11], which can significantly enhance or repress splice site usage. Splicing regulatory proteins can be divided into two major families, differing in their position-dependent effect on splice site usage [12]. Serine- and arginine-rich proteins (SR proteins) enhance usage of upstream splice acceptors and downstream splice donors, but repress usage of upstream splice donors and downstream splice acceptors. Heterogeneous nuclear ribonucleoproteins (hnRNP) on the other hand have an opposite effect on splice site usage. The proximal splice donor context beneficial for its usage therefore consists of upstream binding motifs of SR proteins and downstream binding motifs of hnRNP proteins (reviewed in [11]).

In general, different nucleotide sequences coding for the same amino acid sequence can contain different splicing regulatory elements—binding sites for SR- or hnRNP proteins. For any genomic sequence, its splicing regulatory properties are reflected by its HEXplorer score (HZ_{EI}) profile, calculated for every nucleotide [13]. Here, we examine the possible variation in the total HEXplorer score while preserving the encoded amino acid sequence for a given reading frame. To this end, we designed an algorithm to maximize or minimize total HZ_{EI} by variation of admissible codons. With an average codon degeneracy of three (range 1–6), the number of alternatively admissible nucleotide sequences for a given sequence of N amino acids is $\sim 3^N$ and grows exponentially with the number of codons, N. An exhaustive search in the space of all alternative sequences is therefore very time-consuming and not feasible in practice.

Stochastic optimization algorithms like Monte Carlo or evolutionary algorithms are particularly well suited for optimizing an objective function—total HZ_{EI}—in an exponentially large configuration space ($\sim 3^N$) under a set of constraints (amino acids). Here, we designed a genetic algorithm with recombination between mating configuration populations using crossover, insertion and random mutations, and combined it with a heuristic sliding window approach. This *ModCon* algorithm (Modulator of Context) permits enhancing or silencing splice site usage by manipulating their sequence neighborhoods while preserving the encoded amino acids. As a proof of principle, we applied ModCon to sequences within a splice donor competition reporter and additionally demonstrated the impact of a change in HEXplorer score for a naturally occurring GT site within a common luciferase expression reporter system. We tested the scope of HEXplorer score manipulation with ModCon on a set of 1000 randomly selected human SD sites.

2. Material and methods

2.1. HEXplorer algorithm

The HEXplorer score is based on hexamer frequency differences in 100 nt long neighborhoods upstream compared to downstream of splice donor sites [13], resulting in a Z_{EI}-score for each hexamer. Hexamers predominantly found upstream of splice donor sites have positive Z_{EI}-scores, and they frequently overlap SR protein binding sites, while negative Z_{EI}-score hexamers often relate to hnRNP binding sites. Proceeding from a single hexamer-based quantification to a score for each nucleotide of a genomic sequence, we calculated the HEXplorer score (HZ_{EI}) as the average Z_{EI}-score of all six hexamers overlapping an index nucleotide: $HZ_{EI} = \sum Z_{EI}/6$. The total HZ_{EI} of a sequence stretch, e.g. a splice site neighborhood, indicates its overall splicing regulatory property, likely due to hnRNP or SR protein binding sites. Changes in HEXplorer score induced by mutations have been shown to correlate well with the mutation's impact on nearby splice site usage [14].

For any splice donor site, the overall SRP-mediated impact of its sequence neighborhood on splice site usage is then captured by its *Splice Site HEXplorer Weight* (SSHW), the total upstream minus downstream HZ_{EI} [14]: $SSHW = \sum_{up} HZ_{EI} - \sum_{dn} HZ_{EI}$. The higher the SSHW, the higher the predicted SRP binding potential of a splice site sequence neighborhood, potentially enhancing its usage.

2.2. The ModCon algorithm

To optimize a splice donor site's SSHW, ModCon combines a genetic algorithm applying principles of natural selection and sexual recombination with a sliding window approach, and separately addresses up- and downstream sequences of the given splice site. Driven by the optimization algorithm, ModCon varies the sequence neighborhood of the splice donor site under the constraint of preserving the encoded amino acids and calculates the HEXplorer score of each alternative neighborhood as well as the SSHW.

As input, ModCon takes (1) a coding sequence, (2) the position of the first nucleotide of the "index" GT site within the coding sequence and (3) either maximization or minimization of the target function SSHW. ModCon outputs a coding sequence with a SSHW-optimized alternative neighborhood (16 codons upstream and downstream by default) for the GT site. The graphical abstract provides a structural overview of the ModCon algorithm.

2.2.1. Maximizing or minimizing the total HEXplorer score of a coding sequence

To maximize the SSHW of a splice donor site, ModCon maximizes the total upstream HEXplorer score and minimizes the total downstream HEXplorer score, and *vice versa*.

By default, ModCon considers a sequence window of ±48 nucleotides around the selected index GT site in frame, excluding codons which would overlap with the 11 nucleotides of the GT donor sequence. Synonymous substitutions are then applied to 16 codons ($\triangleq$ 48 nt) upstream and downstream of the GT site to increase or decrease the underlying total HZ_{EI}, possibly regulating GT site usage through introduction or modification of splicing regulatory elements.

Ideally, the total HEXplorer score HZ_{EI} of all sequences encoding the same amino acid sequence would be calculated to determine the highest or lowest total HZ_{EI}. However, a sequence of 16 amino acids can potentially be encoded by up to 6^{16} or 2.8 trillion different nucleotide sequences, because some amino acids can be encoded by up to 6 different codons. Since the HEXplorer score computation of all 2.8 trillion eligible sequences, however, would require extensive time and memory resources, we developed an evolutionary algorithm supplemented by a sliding window approach.

2.2.2. Genetic algorithm

Genetic algorithms can be used to approach optimization tasks with trillions of potential solutions for combinatorial problems by applying principles of genetic recombination and natural selection [15].

Here, a genetic algorithm is developed to combine distinct sets of nucleotide sequences depending on their "fitness", defined by their total HZ_{EI}. It applies a cyclic iterative optimization process that consists of (1) generating an initial sequence population and calculating its fitness, (2) selecting a suitable mating population, (3) creating a new filial generation from it and (4) introducing random mutations (see flowchart Fig. 1).

First, by randomly selecting eligible codons, an initial parental population **F0** of 1000 sequences is generated, all encoding the

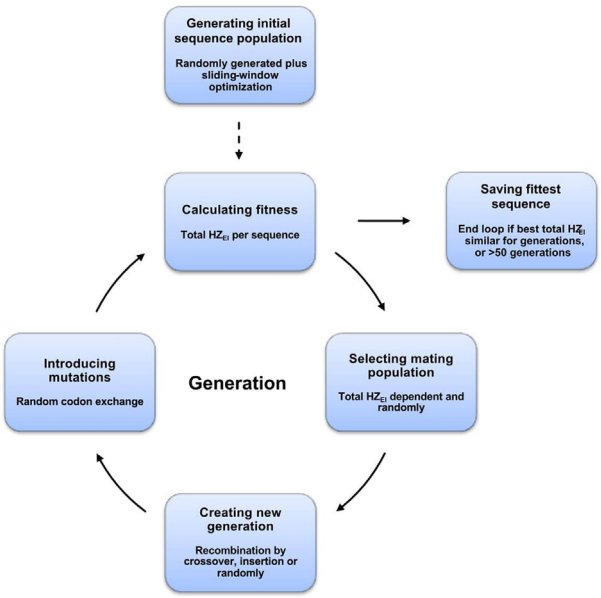

Fig. 1. Schematic of the genetic algorithm. After generation of the initial sequence population, the "fitness" of the sequences, defined by the total HZ_{EI}, is calculated. Next, a defined number of sequences are selected for recombination, either randomly, by crossover, or sequence insertion. Afterwards, every codon within a sequence undergoes random codon exchange with a certain probability. The fitness of every newly generated filial generation of sequences is again determined by total HZ_{EI} calculation. This cycle of generations is continued, while determining the best fitness of every generation, until no further significant increase in total HZ_{EI} can be measured anymore. Finally, the sequence with the highest or lowest total HZ_{EI} is returned. A dashed arrow represents a one-time action, whereas a solid arrow represents an action repeated in every generation.

same initial amino acid sequence. In order to improve convergence speed, this sequence set can optionally be supplemented with typically 100 (10%) sequences previously calculated by a sliding-window approach. This step significantly accelerates the process of the genetic algorithm by directing it towards an optimal solution.

In the first part of the genetic algorithm loop, the fitness of every sequence of the parental population **F0** is calculated as its respective total HZ_{EI}. For each generation, the sequence with the highest or lowest total HZ_{EI} is stored for later reporting on efficiency, and to check if the overall scores could be significantly improved during the last 30 generations.

A new set of sequences, the *mating* population **M**, is then generated through recombination from a subset of the parental population **F0**. This subset is per default assembled from 40% of the fittest sequences of **F0**, 20% of **F0** using the fitness as probability for selection, and 5% randomly selected parental sequences.

Next, **M** is used to generate a set of recombined sequences (300 per default). The resulting *filial* population **F1** is created through random combination of sequence blocks from sequences of the mating population. Every newly generated codon sequence is generated by recombination of two randomly selected codon sequences from the mating population, using three distinct modes of recombination (Fig. 2).

Usage frequency of the three methods of recombination is based on the extent to which the three modes preserve continuous parental codon sequence stretches within the resulting filial sequences. Therefore, 60% of filial sequences originate from "crossover" recombination, where a filial sequence is made from two continuous sequence stretches coming from one parental sequence each. With 30%, the second most applied mode of recombination is "insertion", where filial sequences consist of the sequence of one parental sequence, which holds a random-sized insertion from a second parental sequence in-between. The least used recombination method is the random selection of codons from either one parental sequence. It is used for the remaining 10% of filial sequences.

An important step of evolutionary algorithms is the introduction of mutations after generation of the filial population **F1**, since a carefully selected mutation rate increases the probability to escape potential local maxima or minima during the search for the global peak in the fitness function. From a series of preliminary experiments, we identified an optimal mutation rate of 10^{-4} or 0.01%, meaning that one in 10,000 codons is randomly exchanged with another codon encoding the same amino acid.

The introduction of mutations marks the last step during the cycle of generations. Afterwards, the total HZ_{EI} is again calculated for each sequence, to determine the likeliness to further contribute to the following generations of filial sequences. Then, again, a subset of sequences is selected from **F1** based on their fitness to constitute the next mating sequence population, to generate the second filial population **F2** through recombination.

The generation of newly combined sequences is repeated, until the total HZ_{EI} holds approximately the same level for at least 30 generations or the maximal number of generations (50 generations by default) is reached.

2.2.3. Sliding-window algorithm

In order to keep the computational effort manageable, we furthermore applied a stepwise optimization of codon-quadruplets to optimize the total HZ_{EI} of a 16-codon long sequence. In contrast to the up to 2.8 trillion different nucleotide sequences for a stretch of 16 codons, a four amino acids long sequence can only be encoded by up to $6^4 = 1296$ distinct nucleotide sequences, which enables more efficient total HZ_{EI} calculation.

To optimize the total HZ_{EI} of a sequence, the sliding window algorithm first makes a list of every potential nucleotide sequence encoding the most upstream stretch of four amino acids (codons 1–4). Then, total HZ_{EI} is calculated and the most downstream hexamer of each nucleotide sequence is saved. For every unique hexamer within the sequence pool, the maximal associated total HZ_{EI} is determined. Since the HEXplorer score of each nucleotide is calculated from all six overlapping hexamers, a hexamer between two nucleotides constitutes a barrier in the HZ_{EI} score dependencies. In particular, a sequence downstream of a hexamer can be changed without affecting the HZ_{EI} upstream of that hexamer. The algorithm then proceeds with the optimal nucleotide sequences in each hexamer group, reducing the number of sequences drastically (up to $6^2 = 36$ sequences, in case the last hexamer encodes amino acids with a codon degeneracy of 6).

Subsequently, the algorithm makes a list of every potential nucleotide sequence encoding the next four codons (codons 5–8) and combines every new nucleotide sequence with every previously determined one. Since always those sequences with the highest total HZ_{EI} are selected, the first four codons are now the same in every sequence. To decrease computation time, we can save them for the output and remove them from our sequence list, reducing the sequence length to four codons. This process is repeated until the end of the 16 codons is reached.

While the sliding window algorithm enables fast calculation (taking only a few seconds per run on a standard machine), its fast convergence skips sequences with an intermediate HZ_{EI} increase. The sliding window algorithm is therefore primarily used to quickly obtain a few near-optimal sequences, in particular as supplementary sequences for the genetic algorithm.

2.2.4. Additional sequence processing

A sequence found to maximize or minimize total HZ_{EI} may still contain GT or AG dinucleotides that may accidentally correspond to strong splice donor or acceptor sites. To reduce coincidental

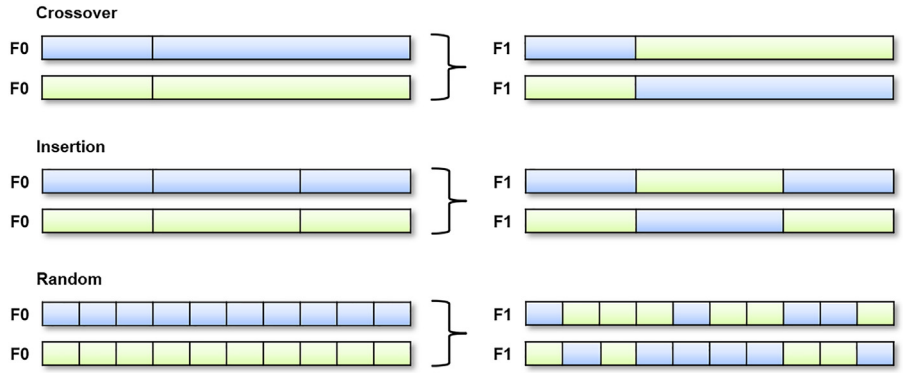

Fig. 2. Modes of sequence recombination. Mating of two codon sequences (F0, F0) of the parental sequence population can lead to various resulting potential filial sequences (F1, F1) depending on the modus of recombination. Filial sequences coming from crossover combinations are constituted of two continuous sequence stretches, coming from one parental codon sequence each. During insertion recombination, the filial sequences consist of one of the parental sequences, holding an insertion from the other one. Random recombination describes the random mixture of codons from both parental sequences.

introduction of such undesired splice sites, ModCon proceeds to degrade splice sites exceeding a threshold of HBS >10 for splice donors and Maxent score >4 for splice acceptors, while preserving the encoded amino acids.

For that purpose, codons overlapping these GT or AG sites are exchanged by alternative codons leading to no or much weaker sites, while keeping total HZ_{EI} close to the identified optimum. Degrading or increasing the HBond score of a specific index GT site of interest can be performed if needed, using the respective R-functions.

2.2.5. HBond, maxent and HEXplorer data sources

Required data for HBond scores of 5′ splice sites [5] and HZ_{EI} scores of splice site neighborhoods [13] were taken from previous work of this group (cf. http://www.uni-duesseldorf.de/rna). The maxent score for human splice sites has been integrated for the evaluation of 3′ splice site strength [7] with kind approval of Gene Yeo. Data for the calculation of 3′ splice site maxent scores were adopted from the website http://hollywood.mit.edu/burge-lab/maxent/download/fordownload/.

2.3. The luciferase reporter

A dual luciferase reporter was used to monitor differences in the splicing outcome upon using ModCon to render an unused 5′ss functional. The reporter construct consists of renilla- and firefly-luciferase transcription units under the control of an SV40 promoter which are terminated by an SV40 polyadenylation site. Renilla luciferase expression was used for normalization of mRNA abundancies in a PCR based readout. For cloning, synonymous mutations were placed in the firefly luciferase coding sequence to create an EcoRV restriction site upstream of an unused splice donor with an HBS of 14. Downstream of the firefly luciferase stop codon, the HIV derived 3′ss $SA7^{opt}$ and an artificial exon (99 bp) were placed [16]. Algorithm amended sequences were inserted as a gene strand synthesized by Eurofins Genomics, Germany (Eurofins Gene Strand #11106560588).

To analyze the splicing pattern, HeLa cells cultivated in Dulbecco's high-glucose modified Eagle's medium (Invitrogen) supplemented with 10% fetal calf serum and 50 μg/ml penicillin and streptomycin each (Invitrogen) were used for transient-transfection experiments. For that, $2.5x10^5$ cells were plated in six-well plates and transfected with 1 μg of the reporter plasmid using TransIT®-LT1 transfection reagent (Mirus Bio LLC US) according to the manufacturer's instructions. Total RNA was isolated 24 h post-transfection by using acid guanidinium thiocyanate-phenol-chloroform. For semiquantitative RT-PCR analyses, RNA was reverse transcribed by using Superscript III reverse transcriptase (Invitrogen) and oligo(dT) primers (Invitrogen) and amplified using the primer pair #6575/#6381, as well as #6167/6168 for the renilla luciferase internal control. Splicing patterns were visualized via a non-denaturing 10% polyacrylamide gel. Primer sequences for the RT-PCR:

#6575 FW (modified) firefly luciferase GTGTTGTTCCATTCCATCACG
#6381 REV firefly luciferase CAGCTGTTCTCCAGCTGT
#6167 FW renilla luciferase GCGTTGATCAAATCTGAAGAAGG
#6168 REV renilla luciferase TTGGACGACGAACTTCACCT

2.4. Splice donor competition reporter

In order to experimentally test the splicing behavior of ModCon designed sequences, 40 nt stretches of either wild type or modified sequence were inserted between two identical copies of a strong 5′ss sequence with an HBond score of 17.5. These two competing 5′ss define the 3′ end of the first exon of an HIV-based two-exon splicing reporter. 40 nt long sequences between the competing 5′ss were derived from FANCA, FANCB and BRCA2 (Suppl. Table 3). All sequences can be obtained upon request.

To analyze the splicing pattern, transient-transfection experiments were carried out as described above. To monitor transfection efficiency, 1 μg of pXGH5 expression plasmid (hGH) was co-transfected. For semiquantitative RT-PCR analyses, RNA was reverse transcribed by using Superscript III reverse transcriptase (Invitrogen) and oligo(dT) primers (Invitrogen) and amplified using the primer pair #3210/#3211, as well as #1224/#1225 for hGH. Splicing patterns were visualized via a non-denaturing 10% polyacrylamide gel. Primer sequences for the RT-PCR:

#3210 TGAGGAGGCTTTTTTGGAGG
#3211 TTCACTAATCGAATGGATCTGTC
#1224 TCTTCCAGCCTCCCATCAGCGTTTGG
#1225 CAACAGAAATCCAACCTAGAGCTGCT

3. Results

3.1. Similar SSHW ranges obtained by GA and SW for 1000 human TSL1 SD sites

In order to determine the achievable range of SSHW optimization, we extracted 185,190 unique splice donors annotated in Ensembl transcripts (version 101) with the highest transcript sup-

port level of 1 (TSL1, Suppl. Table 1). After removing 1% of extremely high and low SSHW values, the remaining 183,339 wild type donor sites (99%) showed SSHW values ranging from around −300 to 1000 (average SSHW 235).

For a random sample of 1000 splice donors drawn from this set (Suppl. Table 2), we then minimized and maximized SSHW using both the sliding window algorithm (SW) and the genetic algorithm with the results from the SW added to the initial population (GA). Table 1 presents the average and standard deviation SSHW difference for the four combinations of algorithm (GA, SW) and optimization (SSHW min/max).

Fig. 3 shows the distributions of minimal and maximal SSHW difference (optimized—wild type) obtained by the GA and SW algorithm. Note that the resulting distributions for the GA and SW algorithm practically coincide, while the maximal SSHW distribution is ~17% narrower and higher compared to the minimal SSHW distribution.

Comparing the distribution of SSHW values achieved with the two algorithms showed no significant differences during SSHW minimization or maximization. However, for around a third of the 1000 SD sites, one of the two approaches performed marginally better. During SSHW minimization and maximization, the GA with the input from the SW algorithm exceeded the achieved SSHW of the SW algorithm alone in 26% of the cases and underperformed for 8% of the SD sites. However, although both algorithms obtained similar extreme values for SSHW, the GA also provides a wide range of intermediate SSHW, and thus permits fine adjustment of potential binding sites for SRPs.

3.2. Faster convergence of GA if SW results are added to initial population

While the sliding window algorithm is deterministic in nature, the genetic algorithm is stochastic and progressively converges to a sequence with optimized SSHW. Convergence speed depends on the choice of mating populations and filial generations, but also on the initial sequence population chosen. Here, we in particular examined the impact of adding 10% sliding-window optimized sequences to the initial generation of the genetic algorithm during total HZ_{EI} maximization. We generated scatterplots of the total HZ_{EI} values of all 300 sequences in each generation both with (Fig. 4B) and without these additional sequences (Fig. 4A). Upon adding 100 SW-optimized sequences generated from the initial WT sequence, the convergence of the genetic algorithm was faster and reached sequences with optimal total HZ_{EI} values with fewer iterations, significantly reducing the algorithm's runtime. These observations also held true for total HZ_{EI} minimizations (data not shown).

In contrast to the sliding window algorithm, the genetic algorithm has the benefit of approaching the total HZ_{EI} maximum or minimum with many slightly different intermediate sequences, and thus avoids getting trapped in local maxima or minima during optimization. This effect could be nicely observed with the example of Fig. 4, where the maximal total HZ_{EI} of the sliding window algorithm was even exceeded after the third generation of the genetic algorithm.

Naturally, the GA with input from the SW algorithm required more CPU time per SSHW minimization or maximization than the SW algorithm alone. During SSHW adjustment of the 1000 human donor sites from Suppl. Table 2 on a machine with 4 CPUs and 8 GB RAM, the SW algorithm took an average of 16.0 s per SD, whereas the GA with input from the SW algorithm took an average of 54.3 s per SD, making the former 3.4 times faster. Similar running time was measured during SSHW minimizations. ModCon, however, also runs with only 1 CPU and a few Mb of RAM available. ModCon per default applies the SW algorithm for SSHW optimizations to save running time, while still achieving a similarly high or low SSHW than with the GA. Alternatively, the combined algorithm can still be applied setting the parameter "optiRate" of the R function "ModCon" to any value other than 100. Setting optiRate to a value >100 results in ModCon using the combined approach to optimize the SSHW of a given GT site. A value lower than 100 triggers the same, but also reports an alternative sequence neighborhood for the donor, which shows optiRate % of the maximal SSHW increase or decrease, enabling fine adjustment of SSHW values.

3.3. Applying ModCon to reporter constructs

To experimentally test the splicing regulatory effect of ModCon designed nucleotide sequences, we selected three 40 nucleotides long wildtype sequences from FANCA, FANCB and BRCA2 with negative HZ_{EI} scores, and positioned them between two identical strong splice donors (HBS 17.5) in an HIV-based two-exon splice site competition reporter. Different sequences placed between these two donor sites can lead to recruitment of splicing regulatory proteins, whose impact on donor usage is position-dependent (Fig. 5A). Whereas hnRNP protein binding enhances upstream donor usage and represses downstream donor usage, SR proteins act in the opposite way.

We specifically selected wildtype sequences with negative HZ_{EI} regions at different levels in order to observe the gradual switch of splice site usage between the competing splice donors. With a HZ_{EI}/nt of −3.28, the wild type FANCB sequence segment induced usage of both donor sites, while a slightly reduced HZ_{EI}/nt of −5.59 in the wild type FANCA sequence segment led to exclusive upstream donor usage, further confirmed by the wild type BRCA2 sequence with HZ_{EI}/nt of −5.90.

Maximizing the total HZ_{EI} of the sequence segments while retaining the amino acid coding information (Fig. 5C) yielded positive HZ_{EI} regions and completely switched splice site usage to the downstream splice donor in all three sequences (Fig. 5B).

In a last step, we examined ModCon on a longer coding sequence with a highly variable HEXplorer profile. Firefly luciferase as a widely used standard expression vector provides a simple experimental read-out. Aiming at turning the firefly luciferase into a splicing reporter, we attempted to switch on usage of a moderately strong GT site (HBS 14 ≈ median HBS = 15 in all human annotated 5′ss, MaxEnt score of 7.33) deep in the coding region that is unused in the wild type luciferase (Fig. 6A).

Using ModCon, we were able to induce usage of the internal, unused GT site by modifying its SSHW from −63.7 to 782.3 and additionally shifting the HEXplorer profile of the sequence seg-

Table 1
SSHW difference, applying the sliding window algorithm (SW) and the genetic algorithm (GA) with the results from the SW added to the initial population. ΔSSHW was calculated subtracting the wild type SSHW from the algorithm achieved SSHW.

	GA ΔSSHW min	SW ΔSSHW min	GA ΔSSHW max.	SW ΔSSHW max
Average SSHW	−1109.6	−1111.8	701.2	701.3
St. Dev. SSHW	227.3	220.8	181.2	181.3

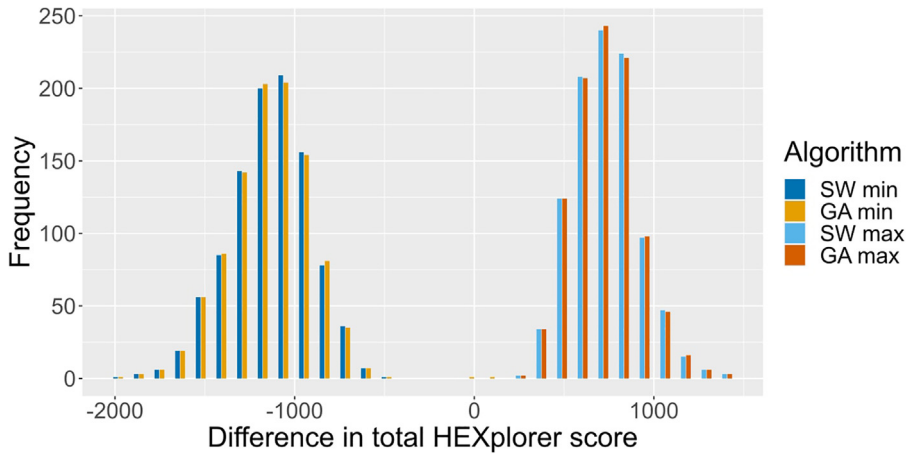

Fig. 3. Bar plot depicting the distribution for the SSHW difference of 1000 human TSL1 SD sites applying different settings of ModCon. SSHW difference (optimized—wild type) is shown on the horizontal axis. Note that bars for GA max and SW max, as well as GA min and SW min lie right next to each other.

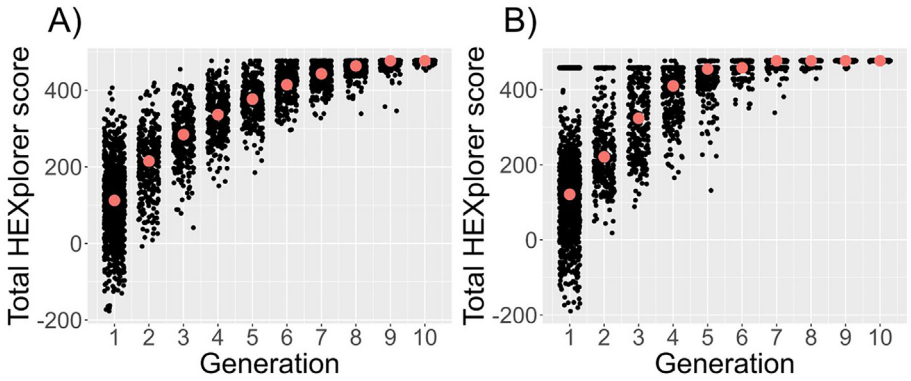

Fig. 4. Total HEXplorer score per generation of the genetic algorithm. Depicted is the total HEXplorer score of every sequence generated during the first 10 generations of the genetic algorithm applied to an exemplary 48 nucleotide long sequence. The first run without spike-in sequences of the sliding window approach shown in (A) needs more generations to reach the maximal total HEXplorer score than with the additional input, shown in (B). For each generation, the median total HEXplorer score is shown in red. Running time on a machine with 4 CPUs and 8 GB RAM for A) 15.3 s and B) 15.7 s. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ment between the GT site and the acceptor site from exon-like to intron-like (Fig. 6B, C). We thus successfully applied ModCon to a much longer and diverse coding sequence.

4. Discussion

Codon degeneracy of amino acid sequences permits an additional "mRNP code" layer underlying the genetic code that is related to RNA processing like pre-mRNA splicing, RNA stability, RNA secondary and tertiary structure, nuclear retention or export [17]. In this study, we addressed pre-mRNA splicing regulation to promote or repress GT site usage without altering encoded amino acids and GT site sequences. Depending on the fine balance between its intrinsic strength and sequence context, GT site usage can be enhanced or repressed by proximal binding of splicing regulatory proteins. Computationally, the neighboring SRP binding landscape is reflected by the SSHW of a given GT site based on its neighborhood's HEXplorer score profile [14]. To systematically modify GT site neighborhoods with respect to their predicted splicing regulatory properties without altering the encoded amino acids, we developed the stochastic optimization algorithm ModCon. We experimentally verified the ModCon algorithm in a splice site competition reporter using wild type sequences from FANCA, FANCB and BRCA2 genes, as well as in a common reporter system of firefly luciferase.

In particular, moderately strong splice sites are most susceptible to regulation by SRP binding. Splicing regulatory proteins binding within a ~50 nt neighborhood of splice donor sites are generally assumed to potentially impact splice site recognition [18,19]. In the evaluation of the ModCon GA and SW algorithm, we therefore used 48 nt wide neighborhoods close to this estimate. The SW algorithm performed equally well as the combined GA during SSHW manipulation, with a 3-times shorter running time. However, the GA allows a much finer SSHW tuning at intermediate levels and avoids local maxima or minima much better than the SW algorithm. Therefore, for SSHW maximization and minimization, ModCon per default applies the SW algorithm and for precise SSHW adjustments, ModCon applies the GA.

Stochastic Monte Carlo or evolutionary algorithms are particularly suited to optimization tasks in large configuration spaces growing exponentially with sequence length, like $\sim 3^N$ for N amino acids in our case. Here, we chose a genetic algorithm with stochastic crossover, insertion and random mutation elements [15]. We successfully reduced computational effort (time and memory demands) by supplementing this genetic algorithm by a heuristic sliding window approach. On a set of 1000 human splice donor sites, both approaches were equally able to significantly optimize SSHW in both directions. In a splice site competition reporter with two identical competing splice donors, we demonstrated that it is possible to induce a switch in SD usage by designing nucleotide

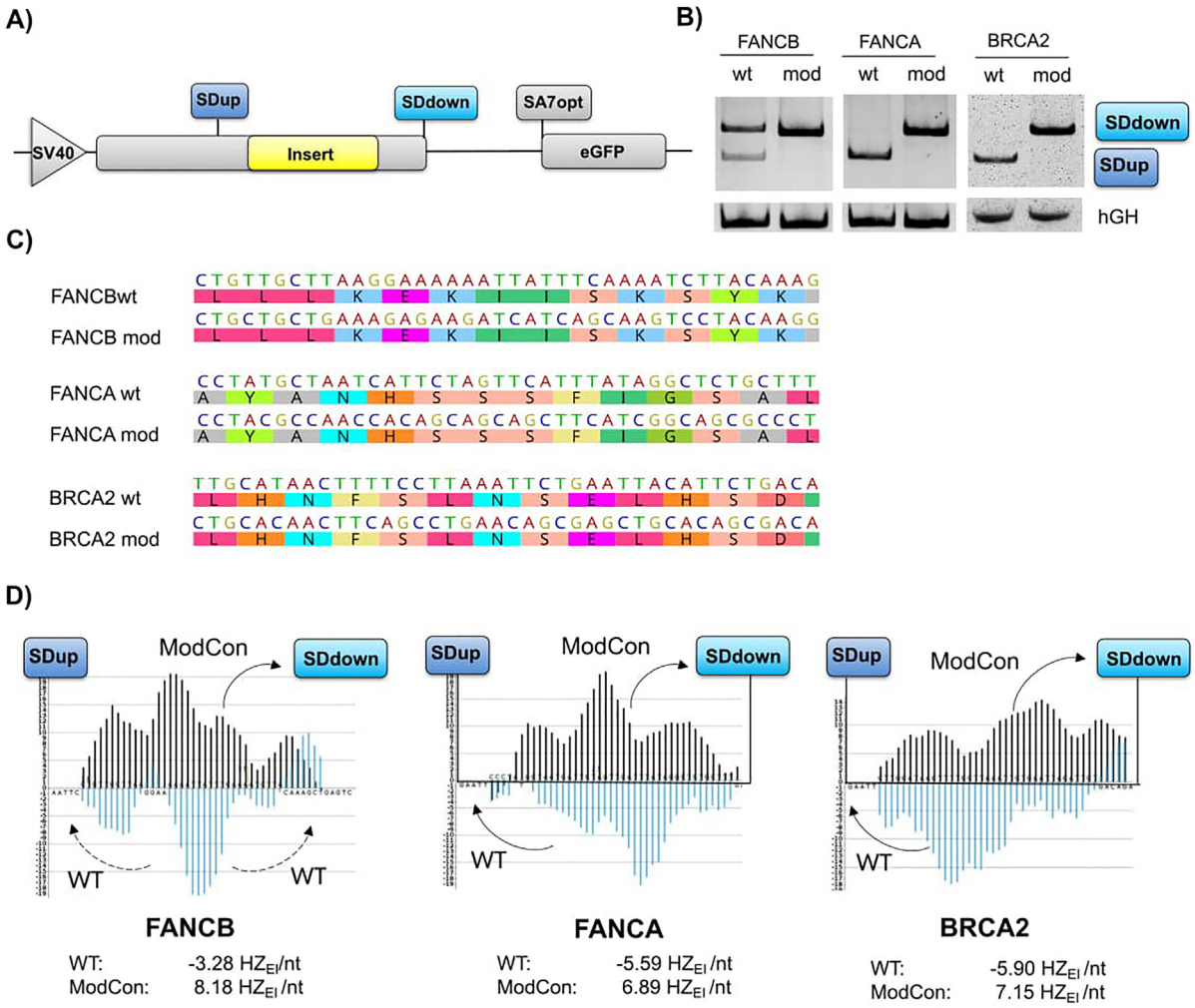

Fig. 5. Switch in splice donor usage by nucleotide sequences encoding the same amino acids. A) Splice donor competition reporter system with the SV40 promoter, the SV40 poly-A site and a strong splice acceptor site. Between the two identical, strong splice donors (SDup and SDdown), any sequence can be inserted and studied regarding its effect on splice donor selection. B) RT-PCR analysis showing a switch in donor usage upon increasing the total HZ_{EI} of the sequence in between, while keeping the coded amino acids, observed for exemplary sequences from FANCB, FANCA or BRCA2. C) Encoded amino acid sequence of the wild type (wt) and the ModCon (mod) generated alternative sequences. D) HEXplorer profiles of the respective tested sequences, with the wild type sequences depicted in blue and the ModCon generated sequences depicted in black. The total HZ_{EI}/nt of the sequences is shown below. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

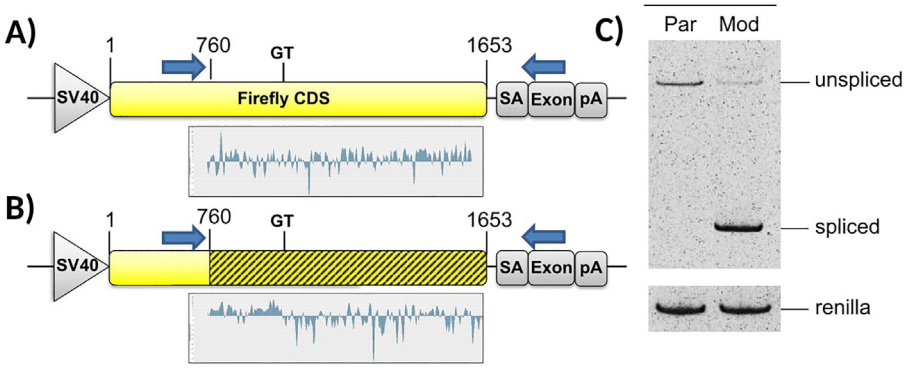

Fig. 6. Luciferase reporter construct encoding identical amino acids and associated splicing pattern. Depicted is the reporter system including the SV40 promoter (**SV40**), a strong splice acceptor site (**SA**), an artificial exon sequence (**Exon**) and the SV40 polyadenylation signal (**pA**). (A) Parental coding sequence of the firefly luciferase, holding an unused GT site (**GT**) at position 1001. (B) ModCon optimized luciferase CDS (hatched), encoding identical amino acids, but containing maximized GT site (**GT**) SSHW and shifted HEXplorer profile between GT site and SA. The 11 nt of the GT site (CAG/GTATCAGG) were not modified. The HEXplorer profiles are shown below the CDS. Primer positions are indicated by blue arrows. (C) RT-PCR analysis showing activation of the internal donor site after modifying the firefly luciferase CDS of the parental construct (**Par**) with ModCon to increase its SSHW (**Mod**). Sequence positions refer to the first nucleotide of the luciferase CDS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

sequences with "opposite sign" HZ_{EI} scores without altering the encoded amino acid sequence. Adjusting the SSHW of an unused GT site within the luciferase reporter enabled activation of this sequence as splice site. To increase the possibility of a splicing event within the CDS and due to position of restriction sites, we modified the 240 nt sequence upstream and additionally adjusted the total HZ_{EI} of the 594 nt long sequence downstream of the GT site to mimic intron-like HZ_{EI} profiles, using functions of the ModCon R-package. All adjustments were done exclusively based on synonymous mutations, maintaining the encoded amino acid sequence. Additional embedded functions allow degradation of the intrinsic strength of cryptic splice sites within a given nucleotide sequence.

The ModCon algorithm and its R package implementation thus open the perspective to conveniently assist in reporter design by either introducing novel splice sites, silencing accidental, undesired splice sites, and by generally modifying the entire mRNP code while maintaining the genetic code.

Availability and implementation

The ModCon R-script is an open source R package available with all needed data in the GitHub repository (https://github.com/caggtaagtat/ModCon). It was uploaded to the Bioconductor R package library.

Author statement

All authors have seen and approved the final version of the manuscript being submitted. They warrant that the article is the authors' original work, hasn't received prior publication and isn't under consideration for publication elsewhere.

Funding

This work was supported by the Forschungskommission of the Medical Faculty, Heinrich-Heine-Universität Düsseldorf (H.S.), the Heinz-Ansmann Stiftung für AIDS-Forschung (H.S.) and the Jürgen Manchot Stiftung (L.M., P.O., A.R., H.S.).

CRediT authorship contribution statement

Johannes Ptok: Conceptualization, Software, Methodology, Investigation, Validation, Visualization, Writing - original draft. **Lisa Müller:** Validation, Visualization, Investigation, Writing - original draft. **Philipp Niklas Ostermann:** Validation, Visualization, Investigation. **Anastasia Richie:** Validation, Visualization, Investigation. **Alexander T. Dilthey:** Methodology. **Stephan Theiss:** Supervision, Writing - review & editing, Writing - original draft. **Heiner Schaal:** Conceptualization, Supervision, Writing - original draft, Writing - review & editing, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank Aljoscha Tersteegen for cloning assistance and Gene Yeo for his friendly approval to integrate the MaxEntScan scoring algorithm into our software.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2021.05.033.

References

[1] Berget SM, Moore C, Sharp PA. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. Proc Natl Acad Sci U S A 1977;74(8):3171–5.
[2] Khodor YL, Rodriguez J, Abruzzi KC, Tang C-H- A, Marr MT, Rosbash M. Nascent-seq indicates widespread cotranscriptional pre-mRNA splicing in Drosophila. Genes Dev 2011;25(23):2502–12.
[3] Aebi M, Hornig H, Padgett RA, Reiser J, Weissmann C. Sequence requirements for splicing of higher eukaryotic nuclear pre-mRNA. Cell 1986;47(4):555–65.
[4] Matera AG, Wang Z. A day in the life of the spliceosome. Nat Rev Mol Cell Biol 2014;15(2):108–21.
[5] Freund M et al. A novel approach to describe a U1 snRNA binding site. Nucleic Acids Res 2003;31(23):6963–75.
[6] Freund M et al. Extended base pair complementarity between U1 snRNA and the 5' splice site does not inhibit splicing in higher eukaryotes, but rather increases 5' splice site recognition. Nucleic Acids Res 2005;33(16):5112–9.
[7] Yeo G, Burge CB. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. J Comput Biol 2004;11(2-3):377–94.
[8] Matlin AJ, Clark F, Smith CWJ. Understanding alternative splicing: towards a cellular code. Nat Rev Mol Cell Biol 2005;6(5):386–98.
[9] Wang Z, Burge CB. Splicing regulation: from a parts list of regulatory elements to an integrated splicing code. RNA 2008;14(5):802–13.
[10] Baralle M, Baralle FE. The splicing code. Biosystems 2018;164:39–48.
[11] Ptok J, Müller L, Theiss S, Schaal H. Context matters: Regulation of splice donor usage. Biochim Biophys Acta Gene Regul Mech 2019;1862(11-12):194391. https://doi.org/10.1016/j.bbagrm.2019.06.002.
[12] Erkelenz S, Mueller WF, Evans MS, Busch A, Schoneweis K, Hertel KJ, et al. Position-dependent splicing activation and repression by SR and hnRNP proteins rely on common mechanisms. RNA 2013;19(1):96–102.
[13] Erkelenz, S., et al., Genomic HEXploring allows landscaping of novel potential splicing regulatory elements. Nucleic Acids Res, 2014. 42(16): p. 10681-97.
[14] Brillen AL et al. Succession of splicing regulatory elements determines cryptic 5ss functionality. Nucleic Acids Res 2017;45(7):4202–16.
[15] Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control. and artificial intelligence 1992.
[16] Kammler S et al. The strength of the HIV-1 3' splice sites affects Rev function. Retrovirology 2006;3:89.
[17] Gehring NH, Wahle E, Fischer U. Deciphering the mRNP code: RNA-bound determinants of post-transcriptional gene regulation. Trends Biochem Sci 2017;42(5):369–82.
[18] Jaganathan K, Kyriazopoulou Panagiotopoulou S, McRae JF, Darbandi SF, Knowles D, Li YI, et al. Predicting splicing from primary sequence with deep learning. Cell 2019;176(3):535–548.e24.
[19] Zhang XH et al. Sequence information for the splicing of human pre-mRNA identified by support vector machine classification. Genome Res 2003;13(12):2637–50.